



**Universidade de Aveiro**  
**2008**

Departamento de  
Engenharia Mecânica

**Luís Filipe Gomes**  
**da Silva**

**Automação em Ambientes Residenciais**

## Júri

Presidente:	Prof. Doutora Mónica Oliveira, Professora Auxiliar, Departamento de Engenharia Mecânica, Universidade de Aveiro
Arguente:	Prof. Doutor Carlos Carneira, Professor Auxiliar, Instituto Superior Técnico, Universidade Técnica de Lisboa
Orientador:	Prof. Doutor José Paulo Santos, Professor Auxiliar, Departamento de Engenharia Mecânica, Universidade de Aveiro
Co-orientador:	Prof. Doutor Rui Moreira, Professor Auxiliar, Departamento de Engenharia Mecânica, Universidade de Aveiro

## **Agradecimentos**

Aos meus pais e à minha família por terem permitido chegar aqui.

Aos meus orientadores, Professor José Paulo Santos e Professor Rui Moreira pela orientação, apoio e compreensão.

Ao Marcos e ao Peter pelo apoio e companhia nos estudos, e aos meus amigos pessoais pela motivação, apoio e paciência.

## Palavras-chave

Domótica, X10, EIB/KNX, Qbus, autómato programável, comunicação, monitorização e controlo remoto.

## Resumo

A automação é hoje uma realidade emergente em ambientes residenciais. O conceito de habitação tem mudado na última década em função da rápida evolução tecnológica. Neste contexto surge a domótica, que tem por objectivo a automatização das habitações. Actualmente, existe uma oferta bastante diferenciada de soluções na área da automação residencial, mas existem limitações ao nível da interligação e da partilha de informação entre diversos sistemas funcionais.

O trabalho desenvolvido demonstra a aplicabilidade de mecanismos de integração de sistemas de domótica. Foram assim desenvolvidas ferramentas de *software* que permitem o controlo e a monitorização, individual e integrada, da rede X10, da rede Qbus e uma rede de autómatos programáveis.

Foram também desenvolvidas interfaces de utilização simples e amigável, que permitem o controlo e monitorização local e remoto do sistema protótipo desenvolvido.

**Key words**

Home automation, X10, EIB/KNX, Qbus, PLC, communication, remote monitoring and control.

**Abstract**

The automation is today an emergent reality in residential environments. The habitation concept has changed in the last decade following the fast technological evolution. In this context and with the main objective of automating day to day activities, the house automation has been considered to be a needed feature in contemporary houses. Currently, several differentiated solutions for residential automation exist, but limitations to the level of the interconnection and the sharing of information the functional systems is usually a reality.

The developed work demonstrates the applicability of integration in home automation systems. Thus, software tools that allowing the control of X10 and Qbus devices along with a local net of PLC's, were developed.

Also simple and friendly interfaces have been developed, allowing the local and remote control of the proposed prototype system.

## Índice

1 Introdução .....	1
1.1.1 Controlo de Iluminação .....	2
1.1.2 Controlo de acessos .....	3
1.1.3 Sistemas de segurança.....	4
1.1.4 Sistemas de videovigilância .....	6
1.1.5 Sistemas de aquecimento e ar condicionado e ventilação .....	6
1.2 Objectivos.....	7
2 Estado da Arte .....	9
2.1 Sistemas de Domótica.....	9
2.1.1 Protocolo X10 .....	9
2.1.2 Protocolo EIB.....	14
2.1.3 Protocolo QBUS.....	20
2.1.4 Sistema baseado num PLC.....	21
2.1.5 Protocolo LonWorks .....	23
2.1.6 Principais diferenças entre as diversas tecnologias .....	27
2.2 Protocolos de Comunicação .....	31
2.2.1 Modelo de referência OSI .....	31
2.2.2 Protocolo de Comunicação RS-232 .....	33
2.2.2 Protocolo de comunicação RS 485 .....	34
2.2.3 Protocolo de comunicação TCP/IP .....	35
2.2.4 Protocolo de comunicação Modbus .....	39
2.2.5 Protocolo de Comunicação ZIGBEE .....	42
3 Arquitectura Proposta.....	45
3.1 Sistema de comunicações da arquitectura proposta.....	47
3.1.1 Comunicação Autómato – Autómato .....	47
3.1.2 Comunicação PC – PLC .....	54
3.1.3 Comunicação X10 – PC .....	57
3.1.4 Comunicação Qbus - PC.....	59
3.1.5 Comunicação PC – Base de dados.....	61

4 Implementação e análise do desempenho .....	65
4.1 Estrutura do sistema protótipo proposto .....	66
4.1.1 Hardware utilizado .....	66
4.1.2 Software desenvolvido .....	71
4.2 Análise de desempenho .....	92
5 Conclusão .....	95
6 Referencias Bibliográficas Utilizadas.....	97
Anexos.....	99
Anexo 1 .....	100
Anexo 2 .....	101
Anexo 3 .....	102
Anexo 4 .....	103

## Índice de figuras

Figura 1 - Módulos (X10) de calha DIN que permite o controlo da luminosidade [1].	3
Figura 2 - Terminal de controlo de acessos com identificação biométrica por impressão digital [2].	4
Figura 3 - Exemplo de um detector de intrusão [3].	4
Figura 4 - Exemplo de um detector de incêndio [3].	5
Figura 5 - Exemplo de um detector de inundação [3].	5
Figura 6 - Exemplo de um sistema de videovigilância [4].	6
Figura 7 - Envio de sinais binários 1 e 0.	11
Figura 8 - Exemplo da transmissão de um comando, no caso A2 ON.	12
Figura 9 - Esquema tipo de uma rede EIB [7].	14
Figura 10 - Imagem com as diversas formas que a topologia física de um sistema EIB pode assumir [7].	15
Figura 11 - Imagem esquemática de uma estrutura EIB.	16
Figura 12 - Esquema tipo do endereçamento físico de um módulo EIB.	17
Figura 13 - Formato dos endereços de grupo.	17
Figura 14 - Imagem de um cabo BUS [7].	18
Figura 15 - Esquema típico do sistema Cardio [8].	22
Figura 16 - Autómato utilizado num sistema Cardio [8].	23
Figura 17- Rede de controlo LonWorks típica [9].	24
Figura 18 - Constituição de um MAC_PDU [9].	26
Figura 19 - Campo NPDU[9].	26
Figura 20 – Estrutura da mensagem recebida segundo o modelo OSI.	32
Figura 21 – Esquema dos pinos de uma ficha de RS232.	33
Figura 22 – Gráfico que mostra a variação da taxa de transferência (bps) em função do comprimento do cabo Rs 485[10].	35
Figura 23 – cabeçalho TCP.	37
Figura 24- Nó mestre a emitir em modo unicast.	40
Figura 25 - Ilustração do nó mestre a emitir em modo Broadcast.	40
Figura 26- Estrutura de uma mensagem no Modbus.	41
Figura 27 - Representação esquemática da arquitectura proposta.	46
Figura 28 - Rede de autómatos ligados por <i>Ethernet</i> /RS485.	47
Figura 29 - Estrutura de mensagens proposta.	49
Figura 30 - Exemplo de aplicação da estrutura de mensagens proposta, no caso a Leitura da saída digital 5 do autómato definido pela estação 2.	49
Figura 31 - Exemplo de aplicação da estrutura de mensagens proposta, no caso a activação da saída digital 1 do autómato definido pela estação 3.	50
Figura 32 - Esquema do algoritmo usado na interpretação da mensagem.	51
Figura 33 - Algoritmo utilizado para o processo de leitura de uma variável do autómato.	52
Figura 34 - Algoritmo utilizado para o processo de escrita de uma variável no autómato.	53



Figura 35 - Ligação entre o PC e a rede de autómatos .....	54
Figura 36 - Exemplo de um projecto criado no Facon Server.....	55
Figura 37 - Estrutura da Mensagem Proposta.....	56
Figura 38 - Esquema de ligação entre o sistema de domótica X10, um comando RF, e o PC. .	57
Figura 39- Modulo CM 11 com saída <i>USB</i> que permite a ligação entre o PC e o sistema de domótica X10. ....	58
Figura 40- Esquema de comunicação entre uma ordem dada por RF para o controlo de um equipamento Qbus ou de um equipamento ligado a um autómato. ....	59
Figura 41 – Esquema da comunicação PC – Qbus. ....	60
Figura 42- Representação esquemática da ligação entre o Pc/Aplicação Vb e a base de dados. ....	61
Figura 43 - Campos da tabela SQL que controla e monitoriza o sistema de domótico proposto. ....	62
Figura 44 - o módulo de aparelho X10 [13]. ....	66
Figura 45 - interruptor de Calha DIN [13]. ....	67
Figura 46 - receptor de radiofrequência de sinais X10 [13]. ....	67
Figura 47- Módulo CM 11[13].....	68
Figura 48 - Comando de radiofrequência [13]. ....	68
Figura 49 – Modulo ETH02, interface de <i>Ethernet</i> [14].....	69
Figura 50 - Dim04-300 – Modulo <i>Dimmer</i> [14].....	69
Figura 51 - SWI04 – Interruptor Qbus [14]. ....	70
Figura 52 – <i>Autómato Fatek Corp</i> , modelo Fbs-60MC [15]. ....	70
Figura 53– <i>Autómato Fatek Corp</i> , modelo Fbs-20MC [15]. ....	71
Figura 54 - Funcionamento do software quando o utilizador controla ou monitoriza o sistema. 76	
Figura 55 - Esquema que representa o funcionamento do software com controlo por parte do utilizador. ....	77
Figura 56 – Panorama Geral da Interface Gráfica desenvolvida. ....	78
Figura 57 – Menu de funções básicas.....	79
Figura 58 – Menu inteligente.....	80
Figura 59 – Monitorização dinâmico do protótipo. ....	81
Figura 60 – Detecção de intrusão no lado “norte” da habitação. ....	81
Figura 61 – Quadro de Comunicações e de Controlo.....	82
Figura 62 – Menu de temporizações. ....	83
Figura 63 – Aspecto da página Web desenvolvida.....	84
Figura 64 – zonas de controlo e gestão dos diversos módulos do protótipo.....	85
Figura 65 – Janelas desenvolvidas no protótipo que contemplam a segurança. ....	86
Figura 66 – Diferenças entre interface Web e Visual Basic. ....	87
Figura 67 – Menu de Funções “inteligentes” .....	87
Figura 68 - Esquema de funcionamento da aplicação flash desenvolvida. ....	88
Figura 69 - Aspecto da aplicação Flash desenvolvida.....	89
Figura 70 - Exemplo de navegação entre menus. ....	90
Figura 71 – Menu de Iluminação geral na aplicação <i>flash</i> desenvolvida. ....	90
Figura 72 – Menu de controlo remoto da iluminação da sala. ....	91
Figura 73 – Menu das funções inteligentes. ....	91

## Índice de Tabelas

Tabela 1- Lista de exemplos de comandos X10 básicos.....	12
Tabela 2 – Esquema dos pinos de uma ficha RS232.....	33
Tabela 3 - Tempos de reacção do sistema protótipo proposto .....	93

## 1 Introdução

O conceito de habitação tem mudado ao longo dos tempos em função da mudança de estilo de vida da sociedade. A rápida evolução tecnológica das últimas décadas contribuiu para esta mudança de estilo de vida. Esta constante mudança e evolução da sociedade tem se repercutido na forma de utilização da habitação que foi adaptando-se às exigências dos seus ocupantes.

A evolução tecnológica nos últimos anos tem contribuído para o desenvolvimento constante dos sistemas de informação, dos sistemas electrónicos e de comunicações. Esta evolução tecnológica tem trazido para os mercados produtos que num passado recente eram meros objectos de ficção científica e que hoje são uma realidade emergente.

Contudo, ainda subsistem limitações ao nível da interligação dos diversos sistemas funcionais. De facto, verifica-se uma total ausência de partilha de informação, onde cada sistema age com total independência, ocorrendo mesmo redundância das funcionalidades.

A domótica tem por objectivo a automatização das habitações para satisfazer as necessidades dos seus ocupantes, tais como:

**Segurança:** segurança contra intrusão, segurança técnica (sistemas de alarme contra incêndios, inundações, etc);

**Conforto:** automação de funções de rotina, controlo de iluminação, controlo de ventilação e climatização, distribuição de áudio e vídeo;

**Gestão Energética:** controlo e racionalização energéticos, telemanutenção, telegestão;

**Comunicações:** são consideradas as comunicações internas, com o exterior, com o objectivo de integra-las da forma mais eficiente e global.

A palavra domótica resulta da junção da palavra «domus» (casa) com «robótica» (controlo automatizado de algo). A domótica pode emergir em novos contextos. Esta pode prestar um

auxílio importante no apoio a pessoas com deficiências ou pessoas idosas, permitindo assim uma melhor qualidade de vida e uma maior autonomia. A domótica pode estar relacionada com a automatização de funções no auxílio e aviso (lembra toma de medicamentos) ou mesmo monitorizar a actividade no interior da habitação.

Desta forma, a domótica será um conceito que fará cada vez mais sentido. Problemas como a gestão de recursos, ausência de pessoas em casa, envelhecimento da população e problemas de segurança, irão contribuir para uma mudança de imagem da domótica, ainda associada nos dias que correm a um luxo, para uma imagem que se relaciona com a necessidade de uma simples ferramenta imprescindível do quotidiano.

Contudo, os sistemas actuais de domótica como KNX-EIB, QBUS, X10 e outros não dão uma resposta completa às exigências dos seus ocupantes, focando-se essencialmente no controlo de iluminação e de estores, podendo ter algumas interacções com os outros subsistemas de segurança, controlo de acessos e outros sistemas funcionais. Assim, um sistema de domótica, no seu conceito mais amplo, deve ser uma solução integrada, dispondo de diversas tecnologias de suporte.

Estas tecnologias de suporte surgem no mercado como soluções independentes, e permitem desenvolver sistemas funcionais, associados a funções normalmente autónomas, como por exemplo: controlo de iluminação, controlo de acessos, segurança, conforto térmico, entre outros.

### **1.1.1 Controlo de Iluminação**

Todos os sistemas de domótica contemplam o controlo de iluminação e da distribuição eléctrica. Normalmente esta função surge como a função central de um sistema de domótica, razão pela qual é usual confundir domótica com o controlo da iluminação. No mercado existem diversas soluções (Figura 1), nomeadamente as soluções inerentes a cada sistema de domótica (X10,EIB, Qbus, etc). O controlo da iluminação pode ser processado manualmente (através de botão e/ou telecomando), por programação horária, telefonicamente, através de um PC, autonomamente por um nível de luminosidade (sensor crepuscular) ou por movimento em quase todos os sistemas de domótica.



Figura 1 - Módulos (X10) de calha DIN que permite o controlo da luminosidade [1].

### 1.1.2 Controlo de acessos

O controlo de acessos é também uma funcionalidade típica de uma casa inteligente. O controlo de acessos é importante ao nível da segurança uma vez que permite ao dono de casa controlar o acesso às instalações.

Existem sistemas de acesso com leitura de cartões de banda magnética ou com inserção de códigos pessoais, mas os mais utilizados e fiáveis recorrem à identificação biométrica (impressão digital (Figura 2), reconhecimento de voz, de palma de mão, etc). Existem no mercado diversos equipamentos que podem ser controlados remotamente através de um PC, de um autómato ou serem completamente autónomos.

Por questões de segurança, o sistema de controlo de acessos é normalmente instalado de forma autónoma e independente, não existindo comunicação com outros sistemas.



Figura 2 - Terminal de controlo de acessos com identificação biométrica por impressão digital [2]

### 1.1.3 Sistemas de segurança

A função da segurança, quando associada a um sistema de domótica, assegura usualmente dois tipos de intervenção:

- segurança relativa a intrusão, usando detectores, alarmes e outras formas de dissuasão e aviso (Figura 3);
- segurança técnica, como por exemplo, detectores de fugas de gás, detectores de incêndio e detecção de inundação entre outras possibilidades.

No que respeita à detecção de intrusão é usual recorrer-se ao uso de detectores de presença, de movimento, ou de passagem, detectores magnéticos de embutir nas janelas e entradas consideradas de risco, detectores volumétricos, fotocélulas, entre outros equipamentos.



Figura 3 - Exemplo de um detector de intrusão [3].

Por outro lado, no campo da segurança técnica, a detecção de incêndio é normalmente realizada por detectores óptico-térmicos (Figura 4) nas zonas susceptíveis de incêndio, tais como: lareiras, zona da caldeira do aquecimento central e cozinha.



**Figura 4 - Exemplo de um detector de incêndio [3]**

Outra forma de segurança técnica é o detector de inundação, que permite a protecção contra os danos causados por entrada de água e inundações. Um dos detectores de inundação mais simples consiste num sensor com 2 contactos abertos (Figura 5), que é instalado em zonas potencialmente afectadas tais como: caves, garagens subterrâneas, lavandaria, cozinhas, quartos de banho e balneários. Este sensor detecta a passagem eléctrica entre os 2 eléctrodos quando estes estão imersos em água.



**Figura 5 - Exemplo de um detector de inundação [3].**

Os detectores normalmente estão ligados a uma central de alarmes, que permite gerir e monitorizar estas funções de detecção, usando recursos de aviso tais como alarmes sonoros, envio de mensagens telefónicas, ou eventualmente notificações via correio electrónico.

### **1.1.4 Sistemas de videovigilância**

Os sistemas de videovigilância (Figura 6) são actualmente considerados indispensáveis numa habitação inteligente, ou mesmo em qualquer tipo de infra-estrutura onde seja exigida algum nível de segurança. No mercado, e face ao aumento da procura deste tipo de sistemas, existe uma grande oferta e um contínuo desenvolvimento destes produtos e serviços.



**Figura 6** - Exemplo de um sistema de videovigilância [4].

### **1.1.5 Sistemas de aquecimento e ar condicionado e ventilação**

Os sistemas de aquecimento e de ar condicionado são fundamentais para garantir uma climatização adequada no ambiente residencial. Os sensores de temperatura e de humidade podem medir estes parâmetros em diversas divisões da casa e activa ou desactivar de forma independente válvulas e controlos de temperatura, regulando assim a temperatura e procurando um equilíbrio entre o conforto e a gestão eficiente de energia. Na ventilação e no ar condicionado existem sensores de corrente de ar que podem monitorizar o estado do sistema de ar condicionado, que devem fazer parte integrante de um sistema de domótica.



## 1.2 Objectivos

No actual panorama das soluções de domótica disponíveis no mercado existe uma grande diversidade de soluções funcionais, às quais não estão integradas entre si. De facto, a autonomia e a inexistência de interligação, ou de partilha de recursos, é uma constante nos equipamentos de domótica actualmente existentes e instalados nas habitações inteligentes.

A falta de integração de sistemas é em parte justificada pela ausência de padrões normalizados que actuam sobre estes equipamentos e também pelo facto de serem produzidos por fabricantes diferentes. No entanto, a interligação dos diversos sistemas e a partilha de recursos assume uma posição de destaque no desenvolvimento de sistemas de domótica integrados e eficientes.

Neste contexto, é objectivo deste trabalho demonstrar a aplicabilidade de mecanismos de integração de sistemas de domótica, associando e interligando as mais diversas tecnologias de suporte existentes actualmente. Para o efeito é estabelecido um conjunto de requisitos para a solução a propor, nomeadamente:

- deve permitir a integração de alguns protocolos de domótica existentes no mercado, nomeadamente a integração da rede X10, a rede QBUS e de uma rede de autómatos programáveis disponíveis no departamento;
- deve ter elevada flexibilidade de reconfiguração;
- deve permitir a monitorização e o controlo remoto da habitação;
- deve ter uma interface de utilização simples e amigável;
- deve ter uma elevada capacidade de expansão e reconfiguração de tipologia.

## **2 Estado da Arte**

O desenvolvimento de novas soluções passa sempre por um estudo inicial das soluções existentes no mercado. Neste contexto, este capítulo tem por objectivo explicar as principais soluções de domótica existentes. De uma forma simplificada podemos descrever um sistema de domótica como um conjunto de sensores e actuadores ligados em rede. Desta forma, os protocolos de comunicação entre equipamentos num sistema de domótica assumem uma grande importância. Por essa razão, neste capítulo também se fará um levantamento bibliográfico de diversos protocolos de comunicação, além dos principais sistemas de domótica.

### **2.1 Sistemas de Domótica**

No panorama actual existe uma grande diversidade de soluções ao nível da domótica. Neste contexto procedeu-se a um levantamento bibliográfico acerca dos sistemas de domótica mais relevantes no mercado.

#### **2.1.1 Protocolo X10**

O Protocolo X10 é uma das tecnologias com maior abundância de produtos, clientes e fornecedores nos Estados Unidos da América. Inicialmente a tecnologia foi mais requisitada pelos amadores ligados à área de automatização de habitações, sendo mesmo a mais utilizada nas instalações domóticas caseiras.

A tecnologia X-10 foi desenvolvida entre 1976 e 1978 pela empresa Pico Electronics Ltd, em Glenrother, Escócia, com a finalidade de transmitir dados através de uma linha eléctrica de baixa velocidade (50 bps na Europa e 60 bps nos EUA). O nome X10 surgiu porque este foi décimo projecto da referida empresa [5].

A patente foi posteriormente adquirida pela empresa X10, Ltd e mantida até ter expirado em 1997 [3]. O X10 é actualmente um protocolo de domótica aberto, permitindo a qualquer fabricante o desenvolvimento de produtos baseados nesta tecnologia, que utiliza a rede eléctrica existente numa habitação para envio dos diversos comandos de controlo.

### ***2.1.1.1 Camada Física***

A tecnologia X10 utiliza a rede de distribuição de energia eléctrica como o principal meio de comunicação entre os diversos equipamentos. Este é um aspecto chave desta tecnologia e a sua maior vantagem face a outros protocolos de domótica. Esta tecnologia não necessita de nenhum elemento central para o seu funcionamento, dizendo por isso que usa uma arquitectura descentralizada.

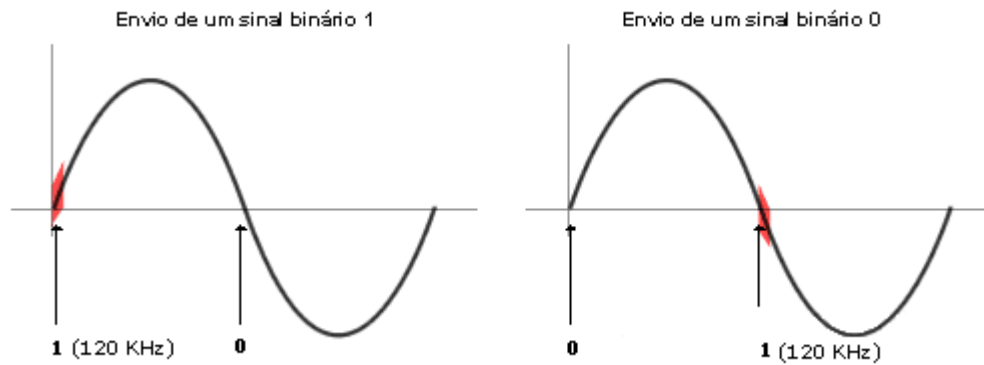
Um sistema X10 pode ser apenas constituído por um conjunto de equipamentos que são directamente controlados pelo utilizador, ou então pode ser constituído por diversos módulos, em que cada módulo consegue receber todos os sinais eléctricos que circulam na rede eléctrica, e o sistema tem de ser capaz de endereçar cada mensagem. Para solucionar a problemática anterior, o protocolo X10 implementou um sistema simples de endereçamento que utiliza 16 códigos do aparelho (1-16) e 16 códigos de casa (A-P), permitindo endereçar sem equívocos 256 aparelhos. A atribuição destes endereços aos vários equipamentos X10 é da responsabilidade de cada utilizador nos próprios equipamentos, actuando sobre dois selectores rotativos, sendo seleccionado em cada um dos selectores um dos códigos referidos anteriormente. Se existirem módulos com o mesmo endereço na nossa rede, esses módulos actuam simultaneamente à mesma ordem (partilha de endereço).

### ***2.1.1.2 Camada Lógica***

A comunicação no protocolo X10 baseia-se na injeção de sinais de alta frequência (120Khz), sobre a rede eléctrica comum, 220V corrente alterna, representando sinais binários (1 ou 0). O sinal é inserido depois da onda sinusoidal de 50Hz ter passado pela origem, com um atraso máximo de 200 micro segundos. Esta particularidade é útil aos receptores para saberem quando devem escutar a linha.

Uma vez que o meio de distribuição de energia é electricamente muito ruidoso, a forma que se encontrou para resolver esse problema, diminuir a probabilidade do ruído eléctrico ser confundido com um sinal válido, foi garantir que um bit nunca é enviado isolado, sendo enviado

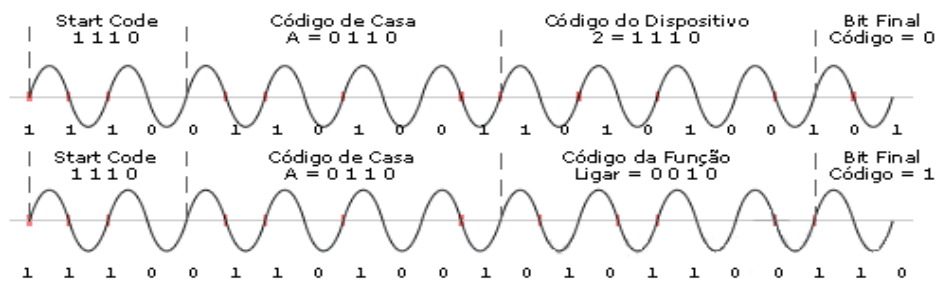
sempre o bit e o seu complemento. Isto significa que o envio do bit 1 é conseguido pelo envio de um bit 1 (sinal de 120kHz na origem) seguido de um 0 (ausência de sinal). No caso do envio de um bit a 0 é enviado um 0 (ausência de sinal) seguido de um 1 (frequência de 120kHz na origem), como ilustrado na Figura 7.[5]



**Figura 7 - Envio de sinais binários 1 e 0.**

No que diz respeito à estrutura das mensagens enviadas neste protocolo, uma transmissão completa de um comando X10 engloba a transmissão de quatro campos que ocupam onze ciclos da onda eléctrica. O primeiro campo representa o *Start Code* – sequência de *bits* (1 1 1 0), e representa 2 ciclos (neste caso não se verifica a regra de cada bit ser enviado de seguida o seu complemento). No campo seguinte figura o código da casa e seus respectivos complementos e é representado em 4 ciclos. De igual forma, seguem-se mais 4 *bits*, que ocupam 4 ciclos, que representa o código do aparelho ou o código da função. Para diferenciar este último campo é enviado um último *bit* (e respectivo complemento), que identifica se o campo anterior se refere ao número de um equipamento (*bit* = 0) ou ao código de uma função (*bit* = 1).

Cada pacote completo deve ser enviado em dois grupos, o primeiro a indicar o aparelho, e o segundo a função a ser executada. Os dois grupos devem ter no máximo três ciclos da onda alternada sinusoidal entre eles. Os comandos *Dim* e *Bright* são excepções a esta regra e devem ser transmitidos continuamente sem nenhum ciclo de intervalo entre eles. Na Figura 8 está descrito um exemplo da transmissão de um comando simples por X10.



**Figura 8 - Exemplo da transmissão de um comando, no caso A2 ON.**

Um comando X10 envolve normalmente duas acções, activar um dado equipamento e de seguida enviar a função a executar [6]. Após a activação de um dado equipamento, ele permanece activo até outro ser referenciado, podendo receber múltiplos comandos, como se pode ver na tabela 1.

**Tabela 1- Lista de exemplos de comandos X10 básicos.**

<b>Lista de Comandos X10</b>		
<b>Código</b>	<b>Função</b>	<b>Descrição</b>
0000	<i>All Units OFF</i>	Desliga todos os equipamentos com o código da casa indicado na mensagem
0001	<i>All Lights ON</i>	Liga todos os equipamentos de iluminação (com capacidade de controlo de brilho)
0010	<i>ON</i>	Liga um equipamento
0011	<i>OFF</i>	Desliga um equipamento
0100	<i>Dim</i>	Reduz a intensidade luminosa
0101	<i>Bright</i>	Aumenta a intensidade luminosa
0111	<i>Extended Code</i>	Código de extensão
1000	<i>Hail Request</i>	Solicita resposta do equipamento com o código da casa indicado na mensagem
1001	<i>Hail Acknowledge</i>	Resposta ao comando anterior
100X	<i>Pré-Set Dim</i>	Permite seleccionar dois níveis pré-definidos de intensidade luminosa
1100	<i>Extended Data</i>	Dados adicionais (seguem-se 8 bytes)
1101	<i>Status is On</i>	Resposta ao pedido Status Request, indicando que o equipamento está ligado
1110	<i>Status is Off</i>	Resposta indicando que o equipamento esta desligado
1111	<i>Status Request</i>	Pedido solicitando o estado de um aparelho

### ***2.1.1.3 Serviços disponíveis no X10***

Com a tecnologia X10 o utilizador está restringido à capacidade de ligar/desligar ou de regular a intensidade luminosa das lâmpadas. É possível fazer o controlo e a monitorização de diversos equipamentos da casa como por exemplo a televisão, as persianas, ou ar condicionado, o sistema de rega. Todavia, nestes equipamentos só é possível utilizar as funções de ligar/desligar.

Existem equipamentos de interface para PC, nomeadamente por ligação RS232, existindo assim diverso software que permite gerir um sistema de domótica com a tecnologia X10. No entanto, é possível para um utilizador avançado criar o seu próprio software, uma vez que o protocolo X10 é um protocolo aberto e amplamente divulgado.

### ***2.1.1.4 Vantagens e Limitações do protocolo X10***

A tecnologia X10 é uma tecnologia antiga e barata e apresenta as seguintes vantagens:

- possui uma atractiva relação custo/benefício;
- constitui uma boa solução para habitações antigas ou já construídas;
- é fácil de instalar e operar.

No entanto, este protocolo possui também um conjunto significativo de limitações que condicionam a sua aplicação generalizada, tais como:

- elevada sensibilidade ao ruído eléctrico;
- reduzida flexibilidade de reconfigurações;
- limitações de endereçamento (no caso da habitação ao lado possuir uma rede X10 poderá haver troca de sinais e confusão no sistema);
- é um protocolo relativamente lento no envio de comandos;
- não suporta a confirmação no envio das mensagens.

## 2.1.2 Protocolo EIB

O EIB é um protocolo de comunicação standard, desenvolvido e orientado pela EIBA (*EIB Association*). Este protocolo está assente sobre a utilização de um barramento de dados (bus) de comando que interliga os vários sistemas envolvidos.

As principais características da instalação eléctrica com bus de comando são as seguintes:

- permite integrar várias funções, automatizando-as de forma independente, mas possibilitando a comunicação entre as mesmas;
- permite manter a flexibilidade de reconfiguração;
- utiliza uma tensão reduzida de segurança, de modo a tornar a instalação ainda mais segura.

### 2.1.2.1 Camada Física

O sistema EIB é caracterizado pela utilização de dois circuitos distintos: um circuito de potência, para a distribuição de alimentação dos respectivos receptores eléctricos, e um circuito de comando para a transmissão de ordens de comando e informações, como se pode visualizar na Figura 9. Esta separação entre o sistema de potência e o de comando além de tornar o sistema fiável, permite em qualquer altura expandir ou modificar facilmente a instalação [7].

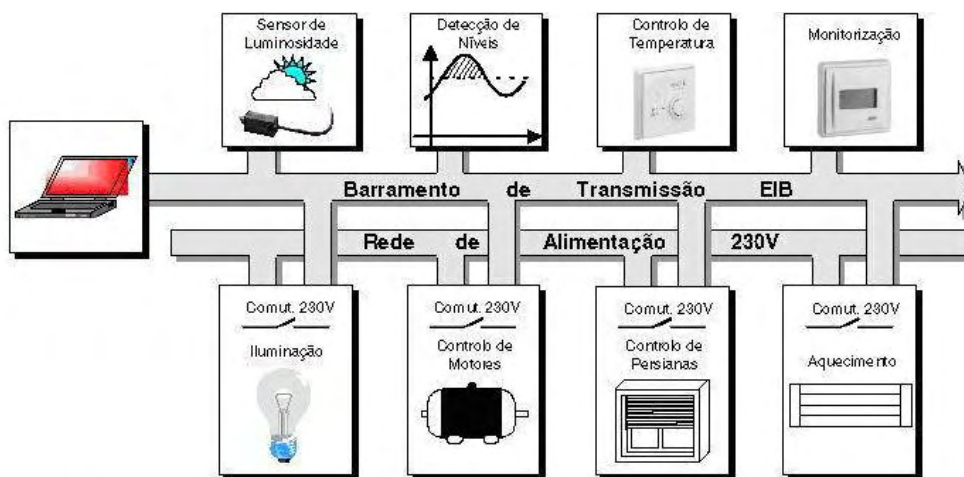
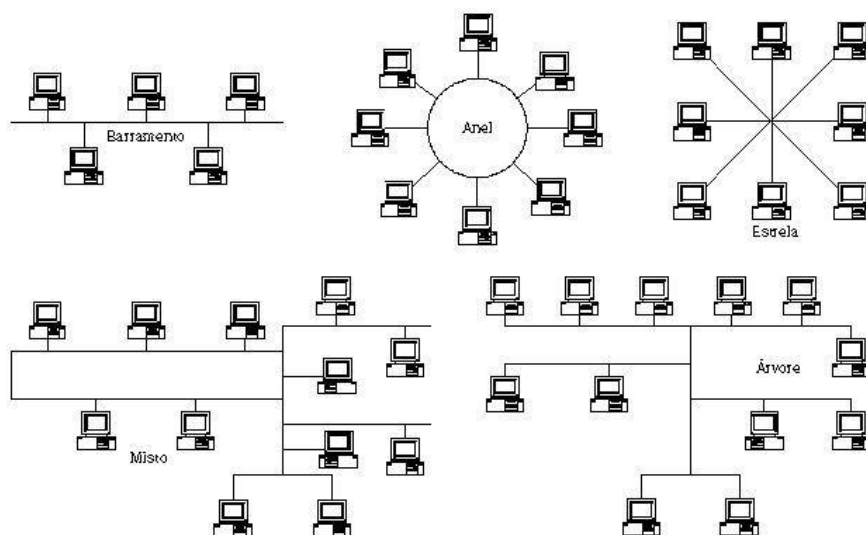


Figura 9 - Esquema tipo de uma rede EIB [7].

A topologia física do EIB pode assumir diversas formas, tais como a topologia linear, em estrela, em anel, em árvore ou uma combinação destas, como se pode visualizar na Figura 10.



**Figura 10 - Imagem com as diversas formas que a topologia física de um sistema EIB pode assumir [7].**

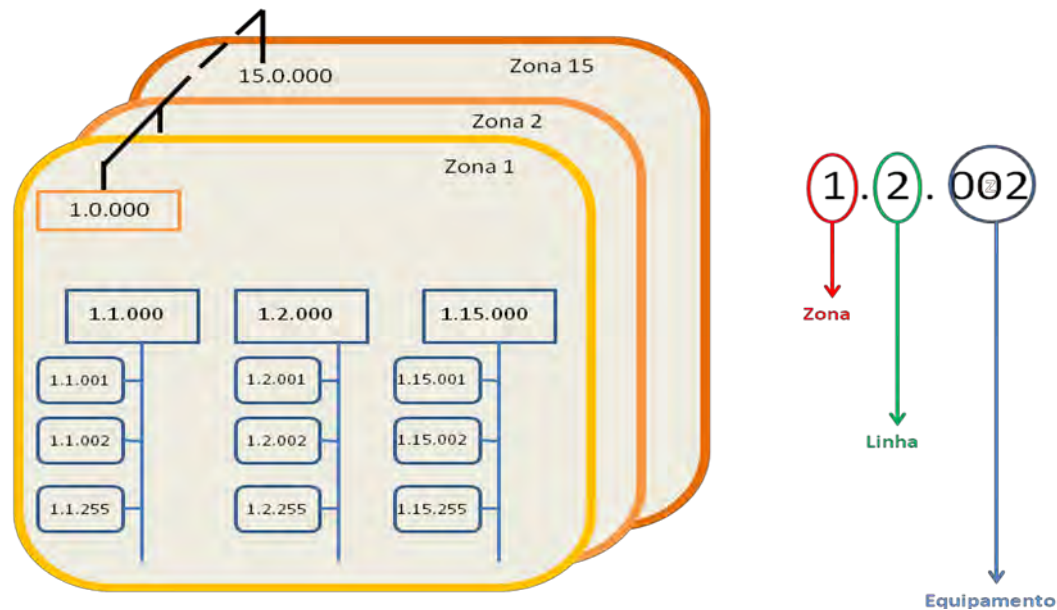
O sistema EIB está organizado segundo uma estrutura hierarquizada, sendo uma rede EIB composta por zonas EIB e estas por linhas EIB.

Uma linha EIB é a entidade mais pequena do sistema, sendo constituída por uma alimentação e por diversos produtos EIB, normalmente designados por participantes. Uma linha EIB suporta no máximo 64 participantes.

Numa zona ou área EIB é possível prever mais de 64 participantes. No entanto, torna-se necessário acrescentar e interligar as várias linhas entre si. Para tal é exigido a definição de uma linha principal, onde todas as outras são ligadas através de acopladores de linha. É possível interligar 15 linhas secundárias a uma linha principal. Ao conjunto de várias linhas dá-se o nome de Zona EIB.

Numa rede EIB de grandes dimensões é possível integrar até um máximo de 15 zonas (Figura 11). Desta forma, a estrutura máxima de uma rede EIB permite interligar e suportar mais de 15000 participantes, dependendo este número do uso ou não de repetidores ao longo do barramento de dados (*bus*) de comando.





**Figura 11 - Imagem esquemática de uma estrutura EIB.**

Existem diferentes tipos de participantes do ponto de vista funcional, os vulgarmente denominados de módulos de entrada ou sensores (botões de pressão, interruptores, detectores de movimento sensores, etc) que são sensíveis a comandos, ordens ou a medição de grandezas físicas, processando e enviando comandos aos módulos de saída. Estes últimos, vulgarmente denominados de módulos de saídas são actuadores que recebem, interpretam e executam as ordens provenientes dos módulos de entrada. Os participantes de saída são na realidade interfaces de potência uma vez que escutam as informações transmitidas pelos módulos de entrada e executam as ordens ou comandos que a eles se destinam.

Numa instalação EIB cada participante inserido no sistema terá uma designação única. Esta designação é conhecida como endereço físico do produto e funciona como Bilhete de Identidade do aparelho. O endereço físico tem uma estrutura estabelecida, como se pode observar na Figura 12.



Existe ainda o endereço de grupo que se define durante a programação da instalação e é utilizado para definir que produtos devem comunicar entre si. Podem existir 2 ou 3 níveis devendo ser escolhido aquele que melhor se adequa à situação. Na Figura 13 pode observar-se uma estrutura tipo dos endereços de grupo de 2 níveis [5].



Os dados são transmitidos através de um par entrançado, podendo também ser transmitidos sobre a rede eléctrica 220V. A transmissão de sinais é feita por meio da diferença de tensão entre os dois condutores do cabo.

17

2x2x0,8 ou J-Y(St)Y 2x2x0,8 são dois exemplos de cabos comuns entre aplicações nesta função e são constituídos por dois pares entrançados:

- par 1: vermelho (+)/ preto (-) -> suporte da comunicação EIB;
- par 2: amarelo/branco -> reserva.

Estes cabos ainda são caracterizados por terem uma bainha exterior em PVC, e apresenta uma blindagem metálica, que reduz o efeito exterior de origem electromagnética. A secção dos condutores é normalmente de 0.8 mm.



**Figura 14 - Imagem de um cabo BUS [7].**

A velocidade média de transmissão de dados é de aproximadamente 9,6kbps. Não é exigida nenhuma combinação de impedâncias.

A informação é modulada sobre baixa tensão (24VDC) e é separada da linha de alimentação dos equipamentos. Tem de existir, pelo menos, uma linha de alimentação por barramento (bus).

### ***2.1.2.3 Serviços disponíveis no EIB***

Numa instalação eléctrica moderna que possua o sistema EIB é possível ter o controlo e a monitorização de diversos domínios tais como:

- controlo de iluminação;
- comando de estores, persianas, toldos ou portões motorizados;
- controlo de aquecimento, ventilação e ar condicionado;
- integração do sistema de segurança;
- comunicação com outros sistemas;
- monitorização do funcionamento da instalação;
- controlo e monitorização da instalação à distância.

O funcionamento do sistema será definido através da programação dos produtos EIB. A programação é feita normalmente através de um PC utilizando um software específico, o ETS (EIB Tool Software), que é adquirido directamente à EIBA. Este programa informático permite a criação do projecto, a escolha dos produtos a utilizar, a atribuição dos endereços físicos, a criação dos endereços de grupo e a parametrização dos produtos.

### ***2.1.2.4 Vantagens e limitações do protocolo EIB***

O protocolo EIB surge como um protocolo com muitas vantagens, razão pela qual é um dos mais utilizados actualmente. As suas principais vantagens são:

- simplificação e redução da cablagem;
- circuito de comando realizado em Tensão Reduzida de Segurança;
- liberdade na atribuição de funções e aplicações;
- integração de comandos à distância;
- agrupamento de comandos;
- facilidade de instalação;
- disponibilização de informações para efeitos de controlo;
- elevada flexibilidade de reconfiguração e expansão.

As suas principais limitações relacionam-se com o elevado preço da sua instalação e a dificuldade associada à programação, que não é acessível ao utilizador comum.

### **2.1.3 Protocolo QBUS**

O Qbus é um protocolo proprietário da empresa Qbus, que desenvolve e comercializa os módulos desta tecnologia.

#### ***2.1.3.1 Camada Física***

O sistema QBUS é um sistema de domótica baseado num conjunto de controladores que se encontram num quadro central ou parcial, as quais se interligam uma série de módulos de entrada e saída, recorrendo a um barramento de dados (*Bus*) de comando com dois condutores.

Os controladores que normalmente se encontram no quadro central têm essencialmente duas funções. A primeira função é a comunicação interna, ou seja o processamento de todos os sinais de entrada e envio da informação a todos os módulos relevantes. A segunda função é a alimentação de todos os módulos QBUS.

Os controladores podem, consoante o seu tipo, adicionar 16, 32 ou 400 saídas distintas. Normalmente cada controlador vem equipado de origem com funções lógicas, capacidade de realizar operações *on line*, entre outras funções.

#### ***2.1.3.2 Camada Lógica***

O barramento de dados (*bus*) do QBUS é caracterizado por ter dois condutores preferencialmente do tipo par entrançado blindado. O barramento de dados (*bus*) tem realmente uma polaridade distinta para cada um dos condutores, no entanto todos os módulos QBUS permitem a troca de polaridade.

O barramento de dados (*bus*) deverá ser preferencialmente instalado numa estrutura tipo anel, mas pode ser igualmente instalado numa estrutura segundo uma topologia em estrela, árvore ou numa combinação entre estas duas variantes. A comunicação pode ser half-duplex ou full-

duplex, ou seja cada módulo pode enviar ou receber sinais para uma unidade central, ou também pode ser simultânea ou seja suporta vários comandos e/ou programações que podem ser efectuadas em simultâneo sem conflitos.

#### ***2.1.3.3 Serviços disponíveis no Qbus***

O QBUS apresenta serviços muito semelhantes ao protocolo EIB, uma vez que utilizam a tecnologia bus como elemento fundamental no seu protocolo. No entanto, o protocolo QBUS é um protocolo proprietário, existindo no site do QBUS determinadas bibliotecas de funções que tornam possível ao utilizador desenvolver *software* de controlo e monitorização de um sistema de domótica.

#### ***2.1.3.4 Vantagens e limitações do protocolo Qbus***

As vantagens do protocolo Qbus são:

- simplificação e redução da cablagem;
- circuito de comando realizado em Tensão Reduzida de Segurança;
- liberdade na atribuição de funções e aplicações;
- integração de comandos à distância.

As principais limitações do protocolo Qbus devem-se ao facto de o protocolo ser proprietário, o que implica uma restrição de soluções disponíveis no mercado e dependência do proprietário.

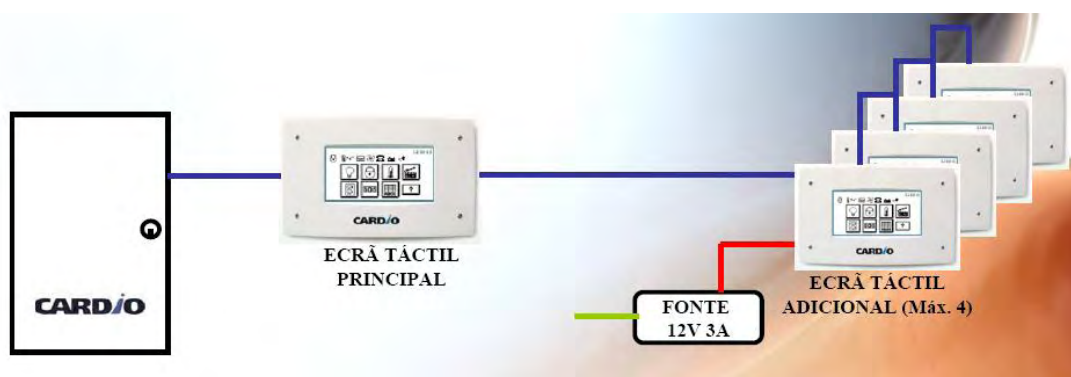
### **2.1.4 Sistema baseado num PLC**

Este sistema de domótica assenta sobre um controlador lógico programável centralizado, ligado directamente aos postos de comando (interruptores, actuadores, sensores).

O sistema Cardio é um exemplo desta forma de domótica, sendo um dos mais utilizados em Portugal, principalmente em habitações do tipo unifamiliar. Este sistema pode facilmente acoplar-se à instalação eléctrica de uma vivenda, tanto em construção, como construída, com algumas modificações relativamente à instalação normal.

Com este sistema podemos controlar praticamente tudo numa habitação, tais como: a iluminação, o ar condicionado, o sistema de rega, a vigilância e segurança entre outros, o que faz com que se tenha o controlo absoluto da casa.

Este sistema é baseado num autómato ligado a uma interface gráfica para o utilizador, que normalmente é uma consola táctil ou variáveis interligadas. Nesta consola é possível controlar, configurar e monitorar todo o sistema de domótica da habitação. A Figura 15 representa de uma forma esquemática um sistema típico de Cardio.



**Figura 15 - Esquema típico do sistema Cardio [8].**

Os autómatos utilizados neste sistema de domótica são autómatos convencionais, ou seja, possuem uma carta de entrada, uma carta de saída e uma porta de ligação *Ethernet*. Na Figura 16 pode visualizar-se um autómato tipo utilizado num sistema de domótica Cardio.

A carta de entradas permite obter informação sobre algo que foi accionado, enquanto que a carta de saída é um mecanismo que permite accionar um determinado mecanismo, por exemplo uma lâmpada ou uma electroválvula. A porta UTP permite estabelecer a comunicação com uma consola táctil central, que pode ou não ter outras emparelhadas. A ligação *Ethernet* permite, por exemplo, ligar o autómato a um PC e este à internet ou a uma rede local, permitindo o controlo remoto através da internet.

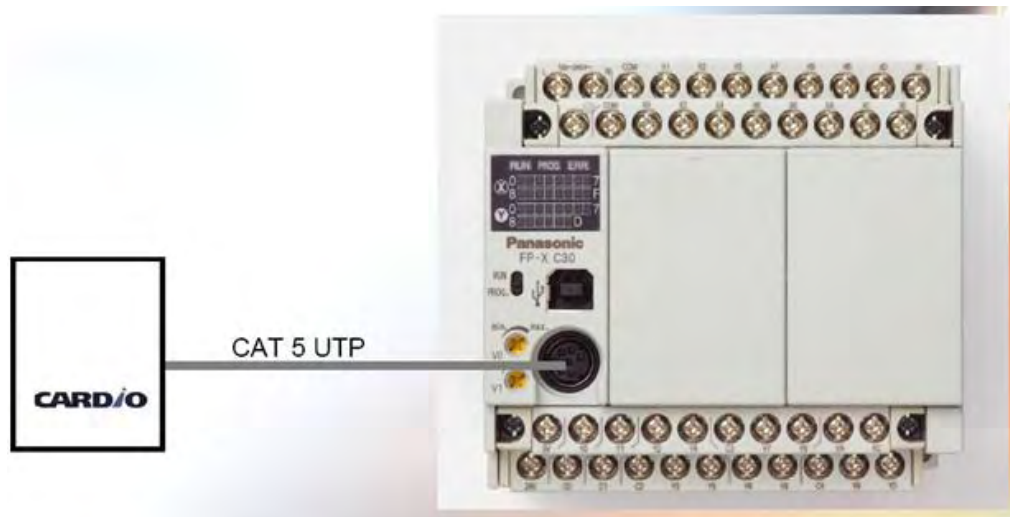


Figura 16 - Autômato utilizado num sistema Cardio [8].

### 2.1.5 Protocolo LonWorks

O sistema *LonWorks* é uma tecnologia que possibilita a criação de sistemas abertos e distribuídos para soluções em automação industrial, residencial, sistemas de transporte e redes de controlo gerais [9].

O sistema *LonWorks* é uma tecnologia que inclui a infra-estrutura de *hardware* e *software* para a operação de uma rede de controlo denominada *Local Operating Network* (LON). Trata-se de uma tecnologia aberta, de controlo distribuído, projectado para implementar a interoperabilidade de redes de controlo.

#### 2.1.5.1 Camada Física

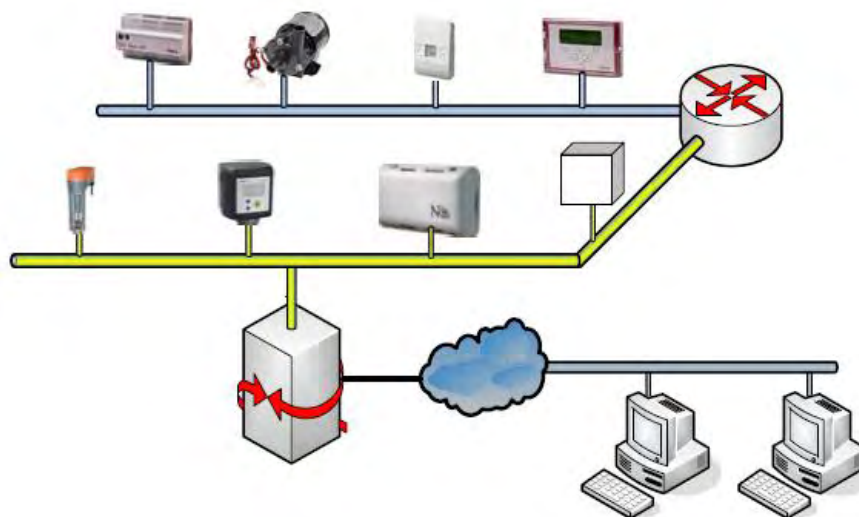
O protocolo de comunicação utilizado é denominado LonTalk (*padrão ANSI – American National Standards Institute – 709.1*) que implementa as sete camadas do modelo de referência OSI (*International Organization for Standardization*). Permite seis meios físicos de transmissão de sinais: cabos eléctricos, par entrançado, cabo coaxial, infravermelho, radiofrequência e fibra óptica. A parte principal de uma rede de controlo LonWorks é um circuito integrado de nome Neuron Chip. Os controladores *Neuron Chip* implementam o protocolo LonTalk (*firmware*) e os *transceivers* que permitem a ligação aos distintos meios físicos de transmissão.



O controlo da rede é distribuído nos equipamentos (nós). Cada nó tem o seu próprio programa, e de acordo com a funcionalidade pretendida, comunica com os outros nós através do protocolo LonTalk.

Os nós presentes no sistema podem dividir-se em dois grupos funcionais: o grupo de nós “controlo”, com memória e capacidade de processamento, e o grupo de nós “transdutores/actuadores”, que actuam unicamente lendo as variáveis dos transdutores e enviando-as através da rede até aos nós de processamento. Similarmente, um nó actuador pode simplesmente activar uma saída em resposta a uma ordem enviada pelo nó de controlo. Esta distribuição é necessária devido ao número limitado de entradas/saídas físicas de cada nó, sendo em muitos casos, impossível integrar num único nó todas as funções de actuação e controlo.

A Figura 17 mostra um exemplo de uma rede de controlo LonWorks típica.



**Figura 17- Rede de controlo LonWorks típica [9].**

O algoritmo de endereçamento no protocolo LonWorks define qual o pacote que é encaminhado de um equipamento para um ou vários equipamentos. Estes pacotes podem ser enviados para um único equipamento, para um qualquer grupo de equipamentos ou para todos os equipamentos.

Para suportar uma rede de equipamentos, pode conter entre dois a dezenas de milhares de equipamentos, o protocolo LonWorks suporta vários tipos de endereços, desde o simples

endereço físico até endereços que designam vários grupos de equipamentos. Os diferentes tipos de endereços são:

- O Endereço físico (*physical address*) onde todos os equipamentos *Lon* contêm um único identificador de 48 bit chamado *Neuron ID*. Este identificador é tipicamente associado na fase de fabrico de um equipamento e nunca sofre alterações até ao fim da sua vida;
- O Endereço de equipamento (*device adress*) onde um equipamento *Lon* é associado um *device adress* quando este equipamento é instalado na rede de equipamentos *Lon*. O *device address* é utilizado porque consegue garantir melhor eficiência na parte do encaminhamento de mensagens e simplifica a reposição de equipamentos com falhas. Este endereço assenta sobre três componentes: o ID de domínio (*domain ID*), o ID da sub rede (*subnet ID*) e o ID do nó (*node ID*); O *domain ID* identifica um conjunto de equipamentos que podem interagir entresi. Pode-se endereçar ter até 32000 equipamentos num domínio. O *subnet ID* identifica um conjunto, que pode ir até 127 equipamentos inseridos num só canal ou num conjunto de canais ligados por repetidores. Este endereço é usado para encaminhar, de forma eficiente, os pacotes na rede. Por domínio pode suportar até 255. O *node ID* identifica o equipamento individual contido numa sub rede;
- O Endereço de grupo é um conjunto lógico de equipamentos inseridos num domínio;
- O Endereço de transmissão tem como responsabilidade a identificação de todos os equipamentos da sub-rede, ou de todos os equipamentos num domínio.

### **2.1.5.2 Camada Lógica**

O formato das mensagens que circulam no protocolo LonWorks pode ser visualizado na Figura 18. Assim a informação transmitida pelo bus é descrita sob a forma de tramas.

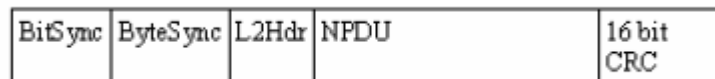


Figura 18 - Constituição de um MAC\_PDU [9].

O *BitSync* e o *ByteSync* permitem que todos os equipamentos sincronizem os seus relógios internos. O campo *L2Hdr* da trama é usado pelo *Mac layer* do protocolo. De seguida é transmitido um campo de dados chamado *Network Protocol Data Unit* ou *NPDU*. A trama é terminada com o envio de um campo de 16 bit CRC que serve para detecção de erros de transmissão. O campo NPDU pode ser dividido nos seguintes semi-campos como se pode observar na Figura 19 [9].

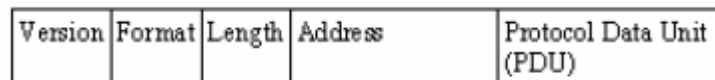


Figura 19 - Campo NPDU[9].

O semi-campo *version* define a versão do protocolo. *Format* define o tipo de formato para o semi-campo *Address* e *Protocol data unit*. Dependendo do semi-campo *format*, o semi-campo *address* pode conter um ou mais dos seguintes endereços: *source address*, *destination address*, *subnet ID*, *Destination SubNet ID*, *Neuron ID*. O *Protocol Data Unit* (PDU) contém os dados actuais para comunicar com os outros equipamentos.

A comunicação obedece ao protocolo CSMA (**C**arrier **S**ense **M**ultiple **A**ccess).

### 2.1.5.3 Vantagens e Desvantagens do Protocolo LonWorks

O sistema LonWorks apresenta-se como uma boa solução a nível técnico, no entanto a sua principal limitação reside no facto da sua quota de mercado na Europa ser reduzida, o que implica limitações a nível de escolha de fornecedores e resultando restrições nas alternativas para a expansão ou manutenção.

O protocolo LonWorks permite:

- utilizar diferentes meios de transmissão;
- criar sistemas de grande dimensão;
- excelentes velocidades de transmissão.

### 2.1.6 Principais diferenças entre as diversas tecnologias

Em seguida serão enunciadas as principais diferenças entre as tecnologias abordadas anteriormente.

Assumiu-se que nesta comparação entre os diversos protocolos não se irá fazer distinção entre o protocolo EIB e o Qbus. A enorme diferença entre eles é que um é o protocolo aberto e outro proprietário. No entanto a nível técnico são muito similares.

Entre os protocolos descritos anteriormente os que tem maior impacto são o X10 e o EIB. O X10 tem uma maior divulgação na América do Norte, enquanto que o EIB tem maior divulgação na Europa. Estes protocolos proporcionam uma maior adesão de novos clientes porque existe no mercado uma grande diversidade de produtos, de fabricantes e de fornecedores, o que permite uma grande variedade de opções e uma fácil dispersão tecnológica.

A dispersão tecnológica associada ao facto de serem ambos protocolos abertos promove a adesão de novos fabricantes e fornecedores, permitindo um aumento constante da sustentabilidade dos protocolos, contribuindo para uma oferta de novos produtos e garantias ao cliente final.

Estabeleceu-se a diferenciação entre os protocolos a diferentes níveis:

- ***Modo de transmissão da informação***

O EIB utiliza um barramento de transmissão próprio (*bus*), com excelentes garantias de fiabilidade de transmissão. Os meios de comunicação são o par trançado ou rede eléctrica. Com o par trançado é possível a instalação de sistemas de grandes dimensões (1000m de comprimento máximo de barramento + 700m de distância máxima entre barramentos; podem ser interligados até 15 barramentos e 15 áreas). O sistema de par trançado é instalado paralelamente com a rede eléctrica.

Se o sistema implementado for uma rede de autómatos, o modo de transmissão pode ter várias opções. Uma rede de autómatos pode comunicar entre si usando as ligações RS 232, *Ethernet*

ou RS 485. Se o modo de comunicação entre a rede de autómatos for o protocolo RS 485 ou *Ethernet*, é possível a instalação de um sistema de grandes dimensões.

No sistema LonWorks o modo de transmissão pode assumir diferentes meios de transmissão: par entrelaçado, rede eléctrica, RF, infravermelhos, fibra óptica e o cabo coaxial.

Actualmente o mais usado é o par entrelaçado ou a rede eléctrica. Permite implementar sistemas de grande dimensão, devido à grande diversidade de endereços, podendo alcançar os 2200m de comprimento máximo do barramento.

O sistema X10 utiliza os condutores de potência da própria rede eléctrica. Este modo de comunicação torna difícil a instalação deste protocolo em edifícios de uso colectivo ou edifícios de elevada complexidade tecnológica. Por outro lado, este modo de comunicação reduz logo à partida os custos da instalação. Este protocolo está limitado em termos de distância a 185 metros, existindo problemas de comunicação para distâncias maiores.

- ***Velocidades de transmissão***

O sistema X10 não ultrapassa os 50-60 bps porque está limitado à frequência da rede eléctrica ser de 50-60 Hz.

O sistema EIB é caracterizado por ter uma velocidade média de transmissão de 9.6 Kbps.

No sistema LonWorks pode atingir-se velocidades máximas de 1,25 Mbps, dependendo do meio utilizado.

Numa rede de autómatos, a velocidade de transmissão depende dos módulos de comunicação utilizados. No caso do módulo de comunicação utilizar o protocolo de comunicação RS 485, pode-se atingir velocidades de transmissão de dados de 921.6 Kbps.

- ***Número máximo de equipamentos***

No protocolo EIB se só forem usados acopladores de linha o sistema chega facilmente a atingir os 1000 equipamentos por área. Se, para além destes também forem usados acopladores de área e repetidores o sistema pode chegar a atingir os 57.600 equipamentos.

No protocolo LonWorks é possível ter um número máximo de 32000 equipamentos enquanto que no protocolo X10 apenas é possível atribuir 256 endereços diferentes a equipamentos.

Numa rede de autómatos é possível endereçar tantos equipamentos quanto o numero de autómatos que a rede possa suportar. Um autómato convencional pode ter 24 saídas digitais e 36 entradas digitais, aos quais poderemos endereçar um ou vários equipamentos, em função da estrutura do nosso sistema.

- ***Alimentação de equipamentos***

No protocolo EIB os equipamentos são alimentados directamente a partir do próprio meio de comunicação. O meio para o envio de dados não é o mesmo meio que alimenta os equipamentos. A informação é modulada e conduzida sobre 24VDC.

No protocolo LonWorks os equipamentos são alimentados a partir de fontes de alimentação próprias existentes em cada nó. Assim, o meio por onde são enviados os dados não é o mesmo que os alimenta.

No protocolo X-10 os equipamentos são alimentados directamente da rede eléctrica, e também é sobre ela que enviam os dados.

Uma rede de autómatos é alimentada pelos convencionais 220VAC, as suas entradas digitais e saídas digitais são alimentadas com 24VDC.

- ***Arquitectura do sistema***

O protocolo EIB opta usualmente pela arquitectura descentralizada. No entanto existe a possibilidade de se optar pela arquitectura centralizada.

O protocolo LonWorks utilizava inicialmente uma arquitectura centralizada. Actualmente e face aos problemas existentes na arquitectura centralizada, utiliza uma arquitectura o mais distribuída possível.

O protocolo X10 é caracterizado por utilizar uma arquitectura descentralizada, enquanto que uma rede de autómatos tem a possibilidade de ter uma arquitectura descentralizada ou centralizada.

- ***Endereçamento de equipamentos e protocolo em geral***

No protocolo EIB pode ser enviada a mensagem para um equipamento, para um grupo de equipamentos, ou para todos os equipamentos. Utiliza um sistema composto por acopladores de linha e de área para expandir a área coberta pelo sistema EIB. Os dados a enviar são modulados e transmitidos sobre as linhas de alimentação (24VDC). O envio de informação obedece ao modelo OSI que é composto por 7 camadas.

O protocolo LonWorks é similar ao sistema EIB neste contexto uma vez que envia a mensagem para um equipamento, para um grupo de equipamentos ou para todos os equipamentos. Utiliza *routers* ou pontes para direccionar as mensagens. A informação é conduzida sobre 42V diferenciais. O envio de informação obedece ao modelo OSI que é composto por 7 camadas.

O protocolo X-10 envia a mensagem para todos os equipamentos do sistema. Só os equipamentos que têm o endereço que a mensagem específica é que a recebem. Transmite os dados sobre a própria rede eléctrica. Não necessita de equipamentos adicionais para distribuir as mensagens a não ser os interruptores contidos nos próprios equipamentos. Apresenta um protocolo de comunicação bastante mais simples e próprio.

Uma rede de autómatos pode comunicar entre si através de diversos meios. A transmissão de dados depende dos módulos de comunicação presentes nos autómatos da rede. A informação é conduzida através dos condutores do meio de comunicação presente na rede, e um autómato pode enviar ou receber mensagens de um ou de vários autómatos, dependendo do protocolo de comunicação ou das mensagens que circulam na rede de autómatos.

## 2.2 Protocolos de Comunicação

Os protocolos de comunicação assumem um papel fundamental num sistema de domótica, uma vez que temos vários equipamentos que comunicam entre si. Desta forma é útil o levantamento bibliográfico de alguns protocolos de comunicação existentes.

### 2.2.1 Modelo de referência OSI

Quando as redes de computadores surgiram, as soluções eram, na maioria das vezes, proprietárias, ou seja uma determinada tecnologia só era suportada pelo fabricante. Uma vez que não havia a possibilidade de misturar soluções de fabricantes diferentes, um mesmo fabricante era responsável por construir praticamente tudo na rede. Para facilitar a interligação de sistemas de computadores a ISO (*International Standards Organization*) desenvolveu um modelo de referência chamado OSI (*Open Systems Interconnection*), para que os fabricantes pudessem criar protocolos a partir desse modelo.

Na Figura 20 encontra-se representada a estrutura de uma mensagem recebida segundo o modelo OSI (este com 7 camadas, uma vez que a camada “Data Link” corresponde à camada de ligação de dados e à camada física) sendo possível identificar os diversos cabeçalhos acrescentados pelas diferentes camadas.





**Figura 20 – Estrutura da mensagem recebida segundo o modelo OSI.**

No que diz respeito à camada de aplicação, esta estabelece a interface entre o protocolo de comunicação e o aplicativo que pediu ou receberá a informação através da rede.

A camada de Apresentação, também chamada camada de Tradução, converte o formato do dado recebido pela camada de Aplicação num formato comum a ser usado na transmissão desse dado, ou seja, um formato entendido pelo protocolo usado.

A camada de Sessão permite que duas aplicações em computadores diferentes estabeleçam uma sessão de comunicação. Nesta sessão, essas aplicações definem como será feita a transmissão de dados e coloca marcações nos dados que estão sendo transmitidos.

A camada de Transporte é responsável por dividir os dados recebidos da camada de sessão em pacotes que serão enviados para a camada de rede. No servidor, esta camada é responsável por receber os pacotes provenientes da camada de rede e agrupa-los num único segmento a ser enviado à camada de Sessão.

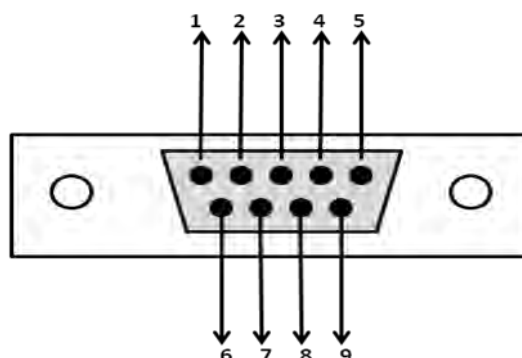
A camada de Rede é responsável pelo endereçamento dos pacotes, convertendo endereços lógicos em endereços físicos (ou IP), de forma que os pacotes consigam chegar correctamente ao destino. Esta camada determina também a rota que os pacotes irão seguir para atingir o destino, baseando-se em factores como condições de tráfego da rede e prioridades.

A camada de Ligação de dados é responsável por identificar os equipamentos numa rede com base nos seus endereços de hardware e, desse modo, controlar o fluxo de dados e organizar os bits da camada Física.

A camada Física define as características técnicas dos equipamentos eléctricos do sistema. Ela contém os equipamentos de comunicação que comunicam directamente com o controlador da camada de Ligação de dados.

### 2.2.2 Protocolo de Comunicação RS-232

O protocolo RS-232 é um dos protocolos de comunicação mais antigos e é uma das formas de estabelecer a comunicação entre dois aparelhos, recorrendo a uma ligação física através de um cabo com diversos condutores, como representa a Figura 21. Este protocolo define as funcionalidades características da camada física do modelo OSI.



**Figura 21 – Esquema dos pinos de uma ficha de RS232.**

**Tabela 2 – Esquema dos pinos de uma ficha RS232.**

Pino	Sinal	Pino	Sinal
1	Detector de dados a enviar	6	Conjunto de dados pronto
2	Recepção de dados	7	Pronto para enviar
3	Transmissão de dados	8	Envio dos dados
4	Terminal de dados pronto	9	Indicador de telefone
5	Terra - Comum		

Para que se estabeleça a ligação é necessária pelo menos a ligação do pino 2 de um aparelho ao pino 3 do outro e uma ligação entre ambos os pinos 5, permitindo apenas para envio de dados (comunicação *simplex*). Existem outros tipos de ligação rs232, *half-duplex* e *full-duplex*. O tipo *full-duplex* permite envio e recepção simultâneo, enquanto o *half-duplex* apenas permite uma operação de cada vez.

No protocolo de comunicação RS-232, os caracteres são enviados um a um como um conjunto de bits. A codificação mais comum é o uso de um *start* bit, seguido por 7 ou 8 bits de dados, opcionalmente um bit de paridade e um ou dois *stop* bits.

### 2.2.2 Protocolo de comunicação RS 485

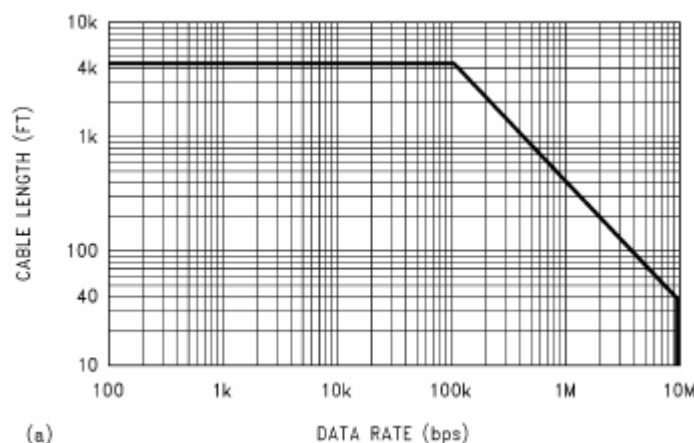
O Protocolo de comunicação RS 485 foi criado em 1985 e é gerido pela *Telecommunication Industry Association* (TIA) que é responsável pelo sector de comunicação da *Electronic Industries Alliance* (EIA), sendo este último credenciado pelo *American National Standards Institute* (ANSI). Este protocolo também corresponde à camada física do modelo OSI.

O protocolo RS-485 recorre a uma comunicação entre 2 ou 4 condutores. O modo de operação é diferencial porque o nível lógico é determinado pela diferença de tensão entre os 2 condutores. Assim, transmite-se um nível lógico 1 quando a tensão no condutor A for positiva e a do condutor B negativa. De igual forma, transmite-se o nível lógico 0 quando B for positivo e A negativo. O protocolo RS 485 é do tipo *half-duplex* ou *full-duplex*. Se a comunicação RS 485 utilizar 2 condutores é *half-duplex*, ou seja ambos podem transmitir e receber dados, mas em diferentes intervalos de tempo. Se a comunicação utilizar 4 condutores é *full-duplex*, ou seja podem enviar e receber dados no mesmo intervalo de tempo.

O protocolo RS-485 utiliza a comunicação multiponto e permite conectar até 32 equipamentos, existindo 1 transmissor e 1 receptor por equipamento.

Uma das vantagens do uso deste protocolo é a sua robustez contra ruído de outros e interferências externas. Outra das características importantes deste protocolo é a capacidade de transmissão a longas distâncias relativamente a outros protocolos, como por exemplo o RS 232. O protocolo Rs485 suporta distâncias até 1200 metros e velocidades de transferência até

10Mbps, no entanto quanto maior for a distância menor será a taxa de transferência, como representado no gráfico da Figura 22.



**Figura 22 – Gráfico que mostra a variação da taxa de transferência (bps) em função do comprimento do cabo Rs 485[10].**

### 2.2.3 Protocolo de comunicação TCP/IP

A internet tal como a conhecemos nos dias de hoje é baseada num conjunto de protocolos de internet criados há bastantes anos atrás com o objectivo de descentralizar as redes existentes na altura e suprir a necessidade de interoperabilidade entre equipamentos. A arquitectura Tcp/Ip surgiu com a criação de uma rede patrocinada pelo departamento de defesa do governo dos USA com o objectivo de manter os órgãos do governo e universidades alertados contra catástrofes que afectassem aquele país. Desta forma surgiu a ARPANET, uma rede que permaneceria intacta caso um dos servidores perdesse ligação. A ARPANET necessitava então de um modelo de protocolos que assegurasse tal funcionalidade esperada, mostrando-se confiável, flexível e de fácil implementação. É então desenvolvida a arquitectura Tcp/Ip. A ARPANET cresceu e tornou-se a rede mundial de computadores a Internet. [11]

### **2.3.3.1 Fundamentos do protocolo IP**

O protocolo IP actua na camada de rede e fornece os serviços de endereçamento e empacotamento tais como:

- identificação de computadores *host* locais ou remotos;
- fragmentação dos pacotes para permitir o seu envio no caso do caminho de destino da rede utilizar um tamanho diferente do pacote;
- eliminação de pacotes antigos (*time-to-live*);
- envio dos pacotes para a camada superior.

Estas acções são reflectidas num cabeçalho acrescentado a mensagem transmitida pela camada inferior [11].

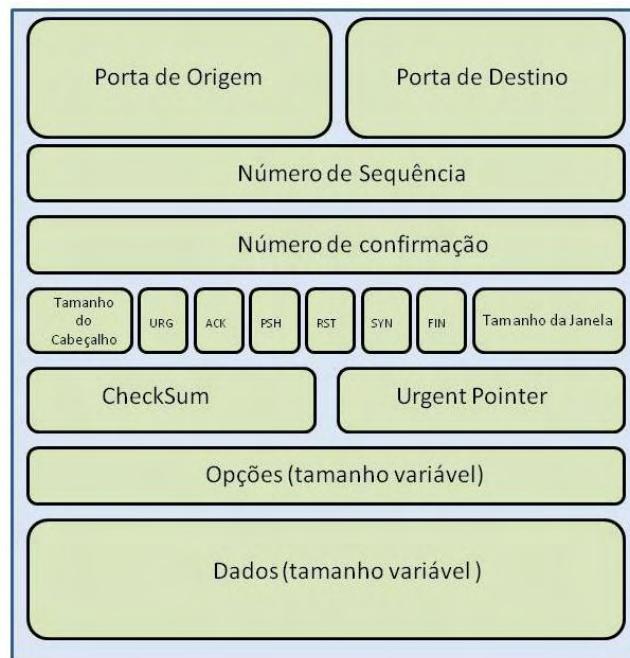
### **2.2.3.2 Fundamentos do protocolo TCP**

O TCP é um protocolo do nível da camada de transporte (camada 4) do Modelo OSI sendo a sua principal função o controlo de fluxo e de erros. As características fundamentais deste protocolo são:

- *full duplex* : é possível a transferência simultânea em ambas direcções (cliente-servidor) durante toda a sessão;
- orientado à *ligação* : a aplicação envia um pedido para o destino e utiliza a ligação física para transferir dados;
- comunicação Extremo a Extremo: *uma ligação* TCP é estabelecida entre o equipamento de origem e o equipamento de destino;
- fiabilidade : o protocolo TCP permite a recuperação de dados corrompidos, eliminação de pacotes duplicados, recuperação de pacotes perdidos e pode recuperar a ligação em caso de problemas no sistema e na rede;
- entrega ordenada – a aplicação faz a entrega ao TCP de blocos de dados com um tamanho arbitrário num fluxo de dados, tipicamente em bytes (também denominados de octetos no contexto de rede de computadores e telecomunicações);
- controlo de fluxo - o receptor, à medida que recebe os dados, envia mensagens ACK (=Acknowledgement), confirmando a recepção de um segmento. O transmissor pode

transmitir segmentos com um número de bytes permitido. Existe a possibilidade de limitar o número de segmentos que circulam entre receptor e transmissor. O TCP garante que, no final da conexão, todos os pacotes foram bem recebidos.

O cabeçalho da mensagem TCP é representado na Figura 23:



**Figura 23 – cabeçalho TCP.**

- O campo “Porta de origem” e “Porta de destino” indicam as portas TCP (16 bits cada) através dos quais vai ser efectuada a transmissão de dados. Por exemplo, a porta 80 é utilizada pelos *browsers* e pelos servidores Web.
- O campo “Numero de Sequência” identifica a posição do segmento no fluxo de dados. Cada transmissão de dados possui um fluxo de dados particular.
- O campo “Numero de confirmação” é utilizado para confirmar a recepção de segmentos enviados anteriormente e especifica o próximo segmento esperado.

- O campo “tamanho do cabeçalho” indica o número de palavras de 32 bits do cabeçalho TCP.
- O campo “tamanho da janela” é utilizado para o controlo de fluxo. Este campo indica ao equipamento emissor quantos fragmentos pode enviar até que o receptor confirme a recepção do primeiro fragmento enviado pelo emissor (*Acknowledge number*). As transmissões de acordo com este protocolo são efectuadas em modo confirmado, isto significa que o receptor tem de confirmar ao emissor o número de fragmentos que recebeu correctamente. Dessa forma, o emissor sabe se necessita de retransmitir algum fragmento.
- O campo “*checksum*” é utilizado para a verificação de erros. O equipamento emissor calcula o checksum da mensagem enviada e armazena-o em 16 *bits*. Esses 16 *bits* são também enviados no final da mensagem. O receptor calcula o *checksum* da mensagem recebida e compara os 16 *bits* por si calculados com os 16 *bits* de *checksum* enviados pelo emissor. Se os dois números forem iguais a mensagem foi correctamente recebida.
- O campo “*urgent pointer*” é utilizado pelo emissor para indicar onde se encontra algum dado urgente dentro do segmento.
- O campo “Opções” é utilizado para a configuração de opções.
- O campo “Dados” contém os dados que de facto se pretendem transmitir.
- O campo “URG” indica que o campo *Urgent Pointer* é válido.
- O campo “ACK” indica que o campo *Acknowledge number* é válido.
- O campo “PSH” é utilizado para indicar ao receptor quando entregar os dados à aplicação imediatamente, sem esperar que o *buffer* fique completo.

- O campo “RST” é utilizado para reiniciar uma transmissão de dados. Este campo também é utilizado para rejeitar uma tentativa de conexão ou rejeitar um segmento inválido.
- O campo “SYN” é utilizado para estabelecer a conexão.
- O campo “FIN” é utilizado para encerrar a transmissão de dados, dizendo que o transmissor não tem mais dados a enviar [11].

### 2.2.4 Protocolo de comunicação Modbus

O Modbus é um protocolo de comunicação de dados criado na década de 1970 pela *Modicon*, e é bastante utilizado em sistemas de automação industrial, principalmente em redes de controladores lógicos programáveis (PLC), para a aquisição de sinais de instrumentos e comandar actuadores.

O protocolo Modbus pode ser descrito como um protocolo actuando segundo um sistema mestre-escravo. Neste sistema existe um nó mestre que envia comandos para o escravo e processa a sua resposta. Os escravos não transmitem dados sem existir uma requisição do nó mestre e nem comunicam com outros escravos. O nó mestre inicia somente uma transacção Modbus de cada vez, podendo estar enlaçados a cada nó mestre 247 escravos.

Os sistemas Modbus podem usar diferentes interfaces físicas, nomeadamente o RS232 ou o RS485. A interface RS485 de 2 fios é actualmente a mais utilizada, sendo possível recorrer à interface RS 232 para a comunicação entre 2 pontos próximos.

O nó mestre pode emitir para um nó escravo uma requisição em dois modos distintos, o modo *Unicast* e o modo Broadcast. O modo *Unicast* (Figura 24) é caracterizado pelo nó mestre endereçar somente um escravo que depois de receber e processar a requisição, envia uma mensagem de resposta para o mestre. Neste modo, uma transacção de Modbus é constituído por duas mensagens: uma requisição do mestre e uma resposta do escravo. Cada escravo deverá ter um endereço único (entre 1 e 247) de forma a poder ser endereçado de forma independente dos outros.



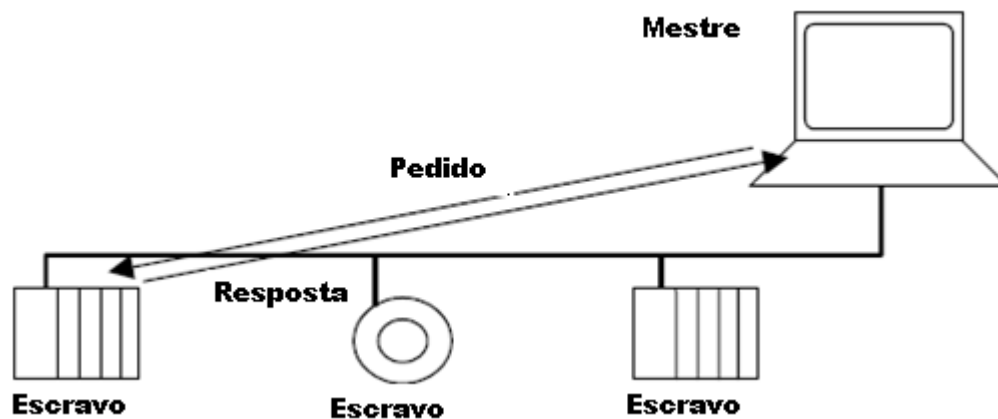


Figura 24- Nó mestre a emitir em modo unicast.

Por outro lado o modo *Broadcast* (Figura 25) é caracterizado pelo facto do nó mestre poder enviar uma mensagem para todos os nós escravos. Neste modo os escravos não respondem uma vez que as requisições de *Broadcast* são necessariamente mensagens de escrita. O endereço 0 é reservado para identificar uma mensagem de Broadcast.

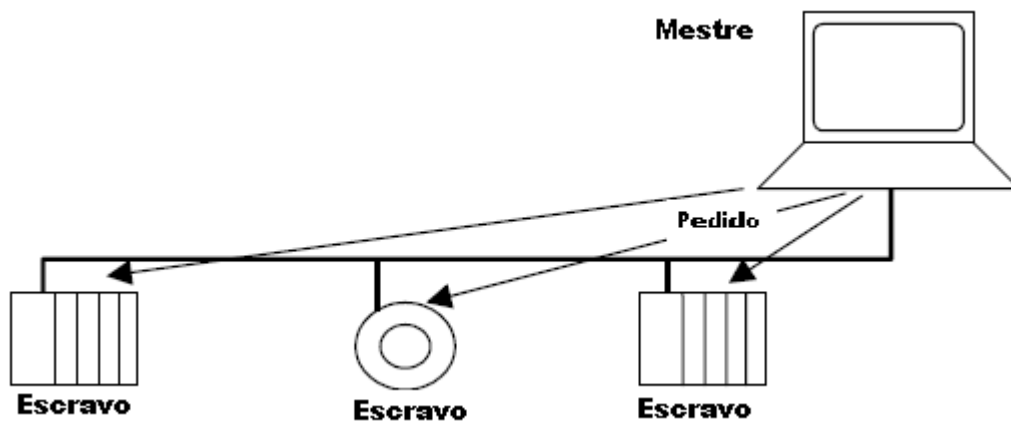


Figura 25 - Ilustração do nó mestre a emitir em modo Broadcast.

O espaço de endereçamento do Modbus compreende 256 caracteres diferentes. O endereço 0 é reservado para a emissão em modo broadcast, e todos os escravos o devem reconhecer como tal. O elemento mestre Modbus não tem endereço específico e somente os nós escravos devem ter um endereço. Os endereços devem ser únicos em cada barramento de Modbus.

#### ***2.2.4.1 Estrutura de Mensagens***

A unidade de dados do protocolo Modbus é constituída por um campo para o código da função, um campo de dados, um campo para o endereço e um campo de verificação de erros (Figura 26).



**Figura 26- Estrutura de uma mensagem no Modbus.**

Numa comunicação Modbus o campo de endereço contém o endereço de um determinado escravo. Um mestre endereça um escravo colocando o endereço do mesmo neste campo. Quando o escravo retorna a resposta, ele coloca o seu próprio endereço no campo de endereçamento.

O código da função indica ao servidor que tipo de ação deve ser executada. Este campo é seguido do campo de dados, onde seguem os parâmetros de requisição e da resposta.

O campo de verificação de erro é resultado da verificação da redundância que é adicionada à mensagem. Existem dois métodos de definição do campo de verificação de erros: RTU e ASCII, cuja aplicação depende do modo de transmissão utilizada.

#### ***2.2.4.2 Meios de Transmissão***

As redes Modbus suportadas pelos protocolos de transmissão em RS232 ou RS485 podem apresentar duas variações na forma de transmissão, o modo RTU e o modo ASCII. O modo de transmissão deve ser comum a todos os elementos ligados à mesma rede Modbus.

No modo RTU os dados são transmitidos no formato binário de 8 bits, o que torna simples a compactação dos dados em pequenos pacotes. No modo RTU os endereços e os valores podem ser representados em binário. No modo ASCII os dados transmitidos são codificados em caracteres ASCII de 7 bits. Apesar de gerarem mensagens facilmente legíveis, este modo consome mais recursos de rede.

#### **2.2.5 Protocolo de Comunicação ZIGBEE**

O protocolo de comunicação Zigbee é uma tecnologia recente, uma vez que a sua primeira versão foi apresentada ao público em Julho de 2005, e surgiu no mercado quando não existia nenhuma norma de redes sem fios globalmente aceite no âmbito de sensores e equipamentos de controlo. Contrapondo-se aos elevados débitos oferecidos por outras tecnologias (como o Bluetooth ou o WiFi), o ZigBee pretende associar a transmissão de dados sem fios a um reduzido consumo energético e uma elevada fiabilidade.

A tecnologia Zigbee tem um vasto potencial de aplicação, desde o controlo industrial à automação de residências (domótica). O protocolo ZigBee possui então determinadas características que o tornam absolutamente diferente dos restantes protocolos. Assim, as suas principais características são:

- reduzido consumo de potência;
- elevada fiabilidade;
- elevada segurança, com recurso a 128 bits de encriptação;
- possibilidade coordenar uma elevada densidade de nós por rede (num máximo de 65535 equipamentos por cada coordenador ZigBee, valor manifestamente superior aos 8 do Bluetooth ou 30 do Wi-Fi);

- admite diferentes topologias da rede: estrela (*star*), malha (*mesh*) ou árvore (*cluster tree*), permitindo o estabelecimento de redes de nós “*ad-hoc*”.

Para além das características enumeradas anteriormente, este protocolo apresenta um tempo de ligação à rede menor que os outros protocolos e apresenta maior rapidez na passagem do modo desligado a activo. Apresenta dois estados de operação: *active* (activo), aquando do envio ou recepção de dados, e *sleep* (inactivo), a aplicação não terá de se preocupar em seleccionar o modo mais adequado.

O protocolo Zigbee permite dois modos de operação da rede: *beaconing* e *non-beaconing*. No modo *beaconing*, os nós ZigBee Routers transmitem periodicamente sinalização (*beacons*) a confirmar a sua presença aos outros nós da rede; os restantes nós só necessitam de estar activos no momento da sinalização. No modo *non-beaconing* os equipamentos mantêm os seus receptores permanentemente activos, existindo um consumo energético superior.

A principal limitação deste protocolo para a aplicação na domótica reside no facto de, apesar de o protocolo estar implementado há cerca de 2 anos, ainda não se encontra consolidado no mercado. De facto, o *hardware* disponível no mercado ainda se encontra em fase de desenvolvimento e os próprios fornecedores destes equipamentos não estão devidamente informados sobre as características e detalhes do equipamento que vendem. Tais factores fazem com que esta tecnologia esteja ainda numa fase muito embrionária, tendo, no entanto, pelas suas características um enorme potencial a médio prazo.

### 3 Arquitectura Proposta

A arquitectura proposta tem como principal objectivo permitir ao utilizador uma grande flexibilidade no sistema de domótica presente na sua habitação. Existem diversas soluções no mercado bastante eficientes e bem implementadas. No entanto, além das limitações inerentes desses sistemas, são sistemas dispendiosos, e principalmente, apresentam uma flexibilidade muito limitada, ou seja não permitem que depois de implementados se possam adaptar-se às necessidades futuras da habitação.

A flexibilidade num sistema de domótica faz todo o sentido na sociedade actual, uma vez que as tendências tecnológicas e sociais justificam a implementação de uma arquitectura flexível. As tendências actuais na área da tecnologia estão relacionadas com a existência de uma maior oferta na área das telecomunicações, um maior aumento da penetração da banda larga, que implica o acesso cada vez mais facilitado em termos económicos a essas tecnologias. Socialmente as tendências estão relacionadas com o envelhecimento da população, com o aumento das famílias unipessoais, com a deslocação de população, com o desaparecimento de famílias numerosas, com a maternidade tardia, entre outros factores que indicam que as

A arquitectura que se propõe neste trabalho prevê a utilização de um autómato, ou uma rede de autómatos, a utilização de um computador pessoal (PC) e a integração deste sistema com outros dois sistemas de domótica, o sistema de domótica X10 e o sistema de domótica Qbus. A Figura 27 representa de forma esquemática a arquitectura proposta.

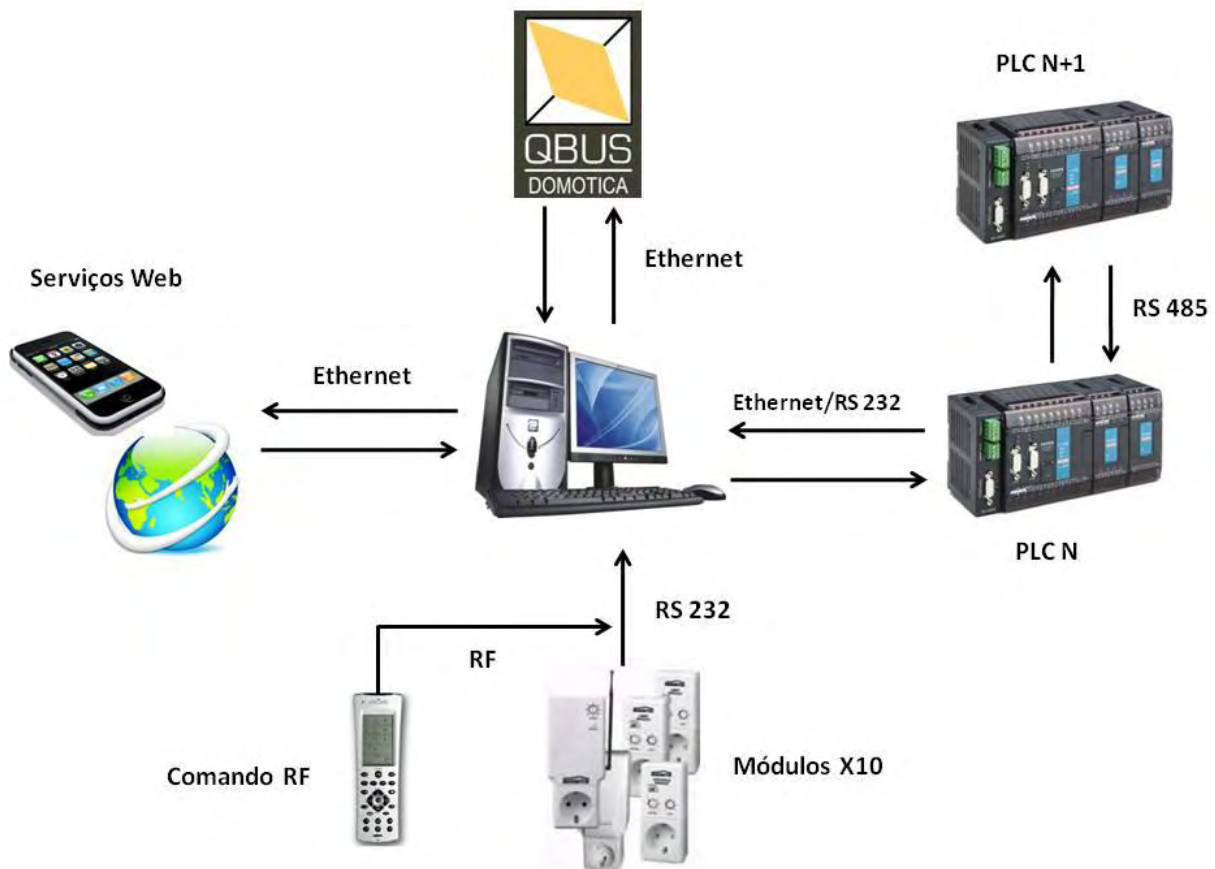


Figura 27 - Representação esquemática da arquitectura proposta.

Na arquitectura proposta os autómatos devem ter a capacidade de controlar sistemas funcionais, como o controlo dos electrodomésticos, da iluminação e dos estores, e deve ser capaz de monitorizar e controlar detectores de incêndio e intrusão, detectores de fugas de água e gás.

O PC surge na arquitectura proposta como membro essencial, uma vez que assume o centro de comando e interligação de sistemas. O PC desempenha um papel fundamental para assegurar a interligação entre o Homem a Máquina e Interface, denominado na domótica como o HMI. O PC garante ainda a capacidade do sistema de domótica possuir a capacidade de ser controlado e monitorizado remotamente através da Web ou através de rádio frequência.

O PC permite a interligação do sistema de domótica a uma base de dados, permite o acesso e a gravação a conteúdos multimédia, pode monitorizar o sistema de videovigilância, alarmes e integrar o sistema de controlo de acessos. Este elemento garante ainda a flexibilização da arquitectura proposta, permitindo a ligação a outros sistemas domóticos (X10,Qbus), quer para a expansão destes sistemas ou apenas para a utilização de algumas funcionalidades.

### 3.1 Sistema de comunicações da arquitectura proposta

As comunicações entre os diversos equipamentos de qualquer sistema de domótica assumem grande importância porque permitem a troca de informação a diversos níveis. Na arquitectura proposta neste trabalho utilizaram-se diversos protocolos de comunicação em função da tecnologia dos equipamentos utilizados.

#### 3.1.1 Comunicação Autómato – Autómato

Os autómatos representam na arquitectura proposta um papel importante uma vez que são responsáveis pelo controlo directo dos sistemas funcionais (Figura 28). Assim, é proposta uma estrutura de mensagens que pode ser usada na ligação entre autómatos, usando o protocolo RS 485 ou o protocolo *Ethernet*.



Figura 28 - Rede de autómatos ligados por *Ethernet*/RS485.

### **3.1.1.1 Estrutura de Mensagem Proposta**

A estrutura de mensagens que se propõe é uma mensagem com 7bytes, como ilustrado na Figura 29 e é constituída por:

Campo Início Mensagem (R + 0): Este campo da mensagem representa o carácter de início de mensagem;

Campo Identifica PLC (R + 1): Este campo representa o endereço do PLC (autómato) para o qual se destina a mensagem;

Campo Função (R + 2): Este campo representa a função a implementar. Pode assumir os valores 0 ou 1. O valor 0 significa que a função que se pretende implementar é um *Read* (Leitura), o valor 1 significa que a função que se pretende implementar ;

Campo Valor da Função (R + 3): Neste campo é definido o valor da função que se pretende implementar, 0 para *OFF*, e 1 para *ON*. Se a função que se pretende implementar for um *Read* (Leitura), neste campo deve ir a identificação do PLC que faz a requisição;

Campo Tipo de Item (R + 4): Este campo define o item que se pretende monitorizar ou controlar: uma saída, uma entrada ou uma posição de memória do PLC. Este campo pode assumir os seguintes valores:

- 1 – entradas digitais;
- 2 – saídas digitais;
- 3 – registos de data;
- 4 – registos;
- 5 – posições de Memória;

Campo Item Número (R + 5): Não basta definir no campo anterior o tipo de item que se pretende monitorizar ou controlar pois existem vários itens do mesmo tipo em cada PLC. Este



campo indica qual o item específico que se pretende controlar o monitorizar. Este campo pode assumir diferentes valores em função do endereço físico em causa;

Campo Fim de Mensagem (R + 6): Este campo da mensagem representa o carácter de fim de mensagem.

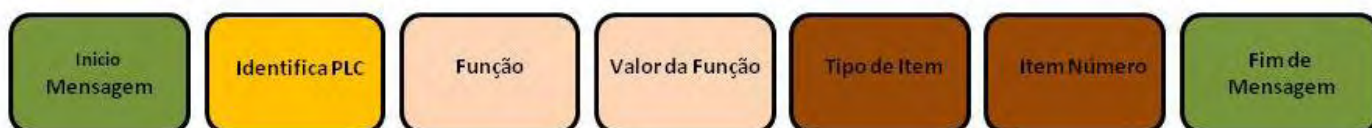


Figura 29 - Estrutura de mensagens proposta

A Figura 30 representa um exemplo de implementação desta estrutura de mensagem. Neste exemplo pretende-se ler a saída digital nº 5 do autómato nº2. Foi definido que os campos de início e de fim de mensagem seriam respectivamente 253 e 254.

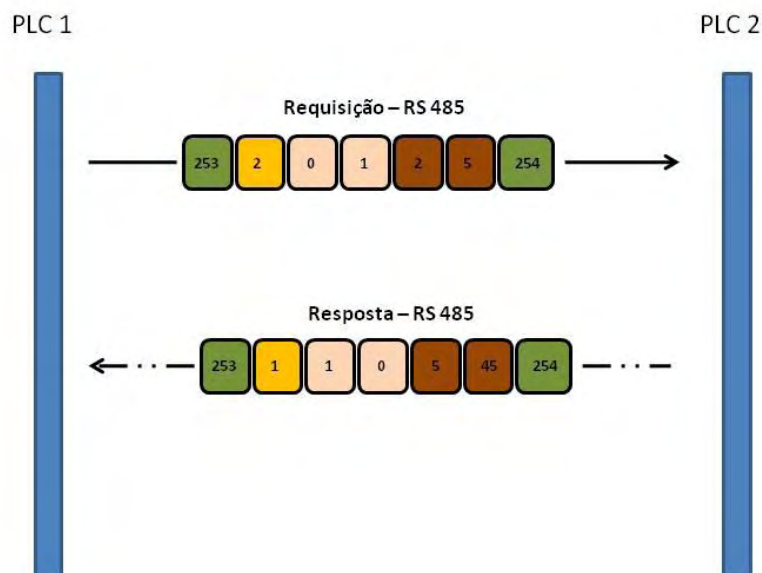
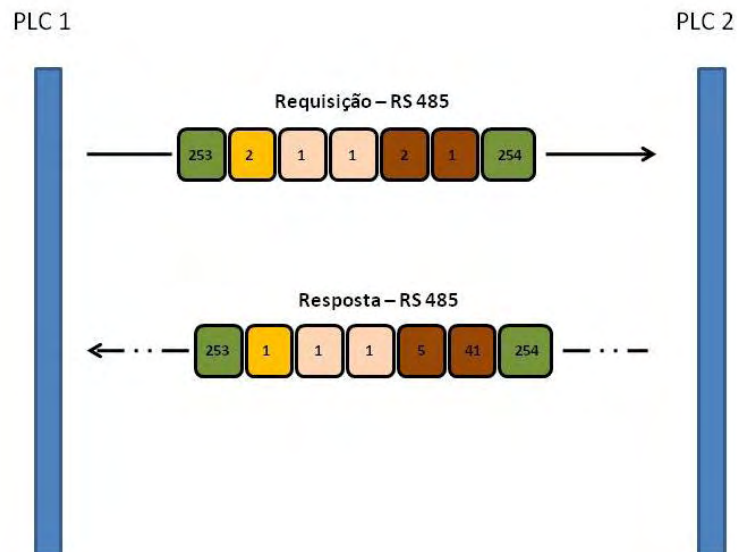


Figura 30 - Exemplo de aplicação da estrutura de mensagens proposta, no caso a Leitura da saída digital 5 do autómato definido pela estação 2.

A Figura 31 representa outro exemplo de implementação onde se pretende activar a saída digital 1 do autómato nº 2. De igual forma, foi definido que os campos de início e fim de mensagem seriam respectivamente 253 e 254.



**Figura 31 - Exemplo de aplicação da estrutura de mensagens proposta, no caso a activação da saída digital 1 do autómato definido pela estação 3.**

### ***3.1.1.2 Interpretação e Construção da Mensagem Proposta em linguagem Ladder***

Depois de definida a estrutura de mensagem houve necessidade de criar um algoritmo de interpretação e construção da mensagem proposta em linguagem *ladder* que é um auxílio gráfico para programação de Controladores Lógicos Programáveis (PLC) no qual as funções lógicas são representadas através de contactos e bobinas, de modo análogo a um esquema eléctrico com os contactos dos transdutores e actuadores. O esquema da Figura 32 representa parte do algoritmo de interpretação da mensagem proposta em linguagem *ladder*.

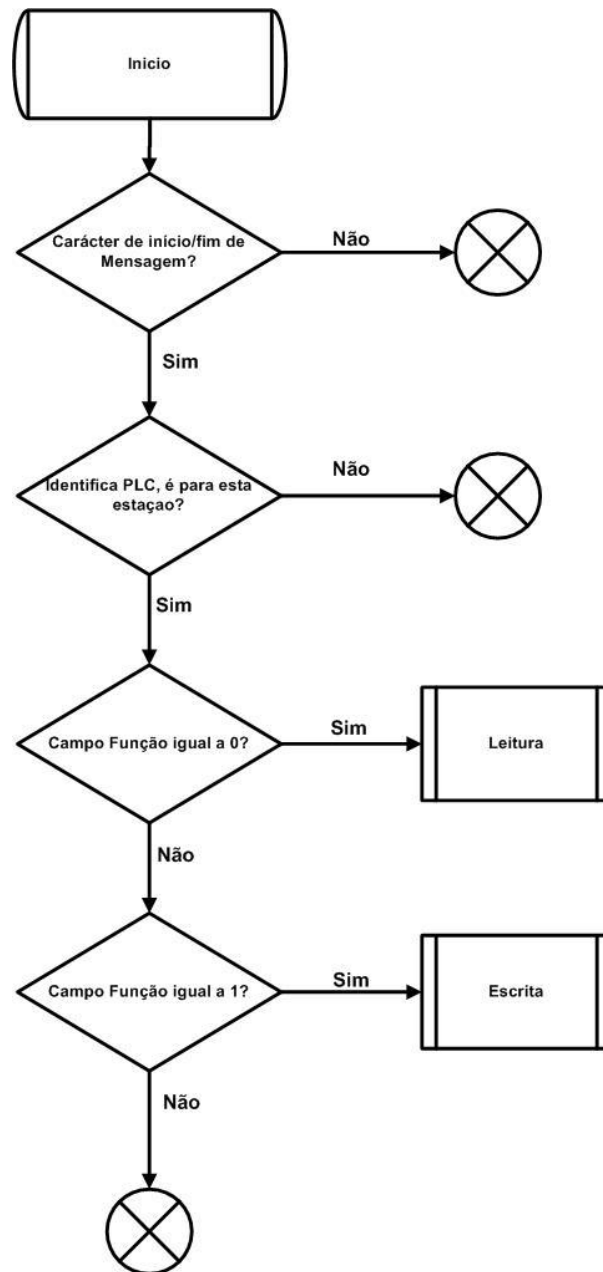


Figura 32 - Esquema do algoritmo usado na interpretação da mensagem

As posições de memória que correspondem à leitura e à escrita de variáveis no autômato são diferentes por motivos óbvios. O algoritmo que seguem, apesar das semelhanças inerentes ao processo de identificação da variável que se pretende ler ou escrever, tem um fim de ciclo diferente. Enquanto na leitura pretende-se a criação de uma mensagem de resposta para o autômato que fez o pedido, na escrita pretende-se a activação de uma variável do autômato.

Nas Figuras 33 e 34 representa-se o algoritmo utilizado para o processo de leitura e escrita de uma variável do autómato, respectivamente.

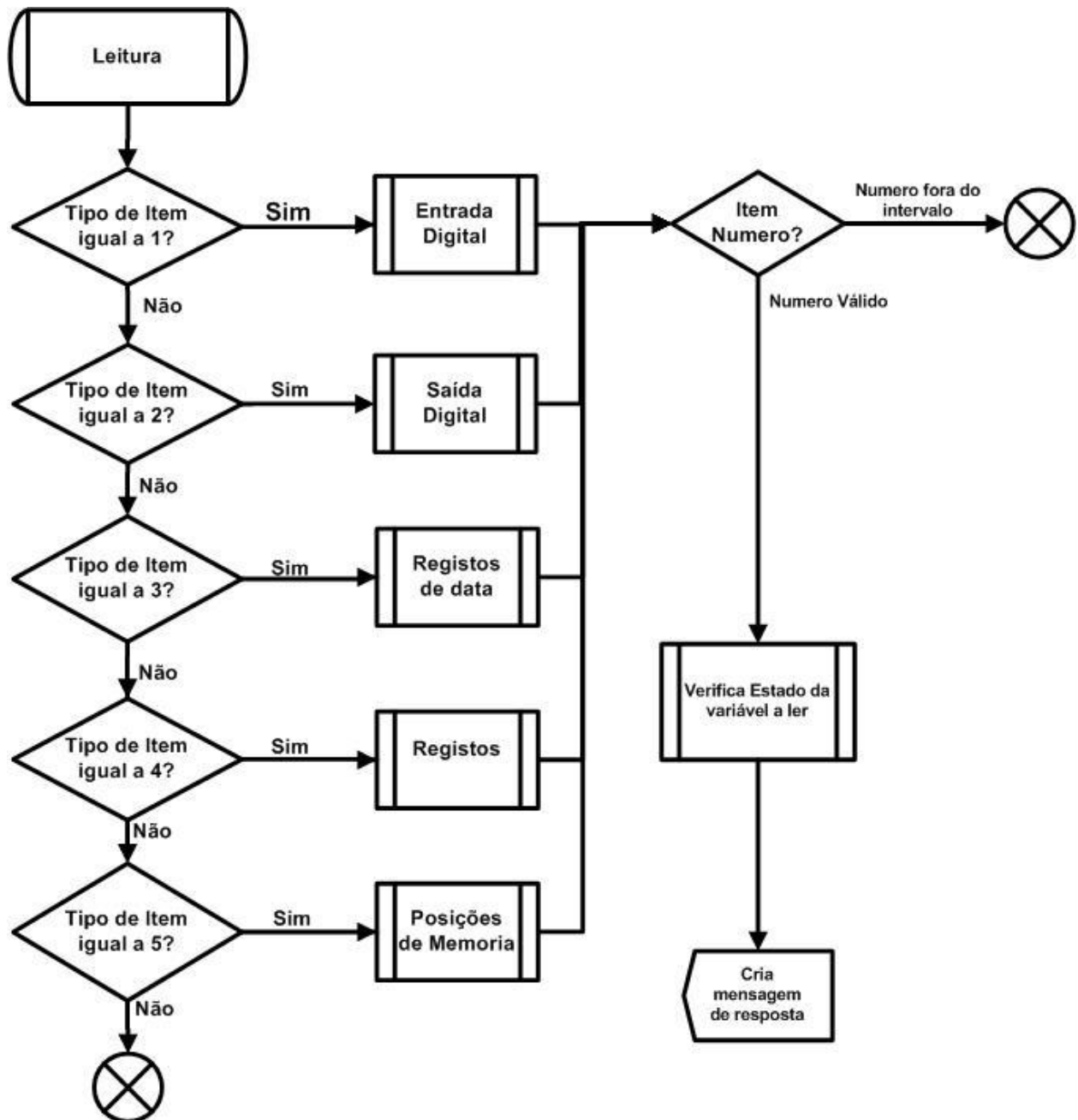


Figura 33 - Algoritmo utilizado para o processo de leitura de uma variável do autómato

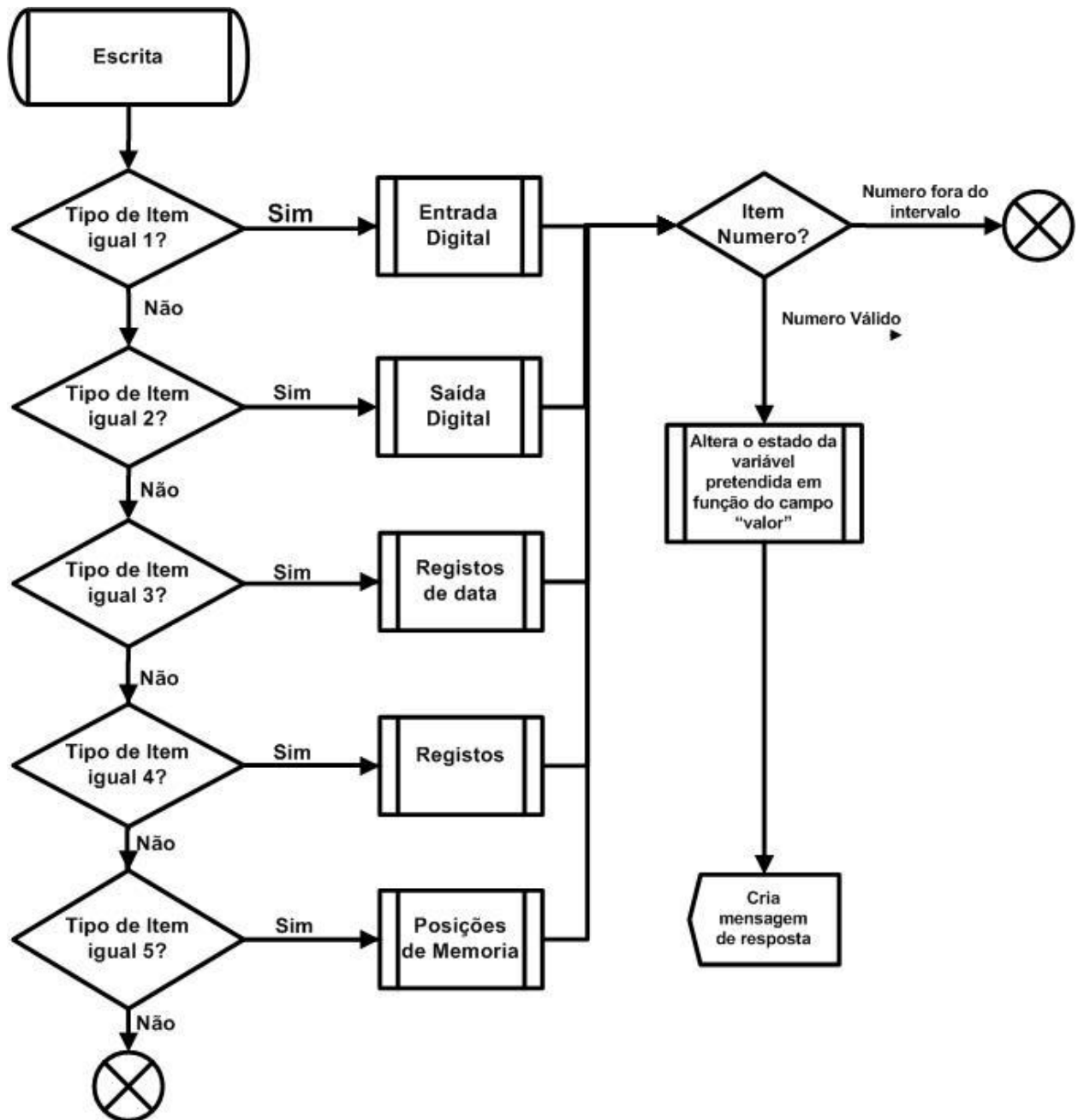


Figura 34 - Algoritmo utilizado para o processo de escrita de uma variável no autômato

O algoritmo implementado para a componente de leitura termina com a elaboração da mensagem de resposta ao autômato que lhe fez o pedido. Esta mensagem de resposta vai estar de acordo com a mensagem proposta, tendo o campo da “função”(R + 2), o valor 1, correspondente à função de escrita; o campo “identifica PLC” (R + 1) para o qual se destina a

mensagem é definido pelo campo “valor da função” (R +3) da mensagem anterior; o campo “valor da função” (campo R + 3) da mensagem actual de resposta, o campo “valor” apresenta o estado/valor que é definido pelo estado da variável que se pretende ler. A informação é guardada numa posição de memória definida para o efeito no autómato que lhe fez o pedido.

O algoritmo implementado para a parte de escrita culmina com a activação ou desactivação da variável do autómato e com a criação da respectiva mensagem de resposta, para que o autómato que lhe fez o pedido saiba o seu estado actual.

### 3.1.2 Comunicação PC – PLC

Na arquitectura proposta a comunicação entre o PC e o PLC pode ser efectuada utilizando a estrutura de mensagens descritas anteriormente. Alternativamente pode recorrer-se a um protocolo específico da *Fatek Automation Corp.* Assim perante estas duas opções é possível comunicar com a rede de autómatos através de *Ethernet* ou RS 232. A Figura 35 representa a ligação entre o PC e a rede de autómatos.



Figura 35 - Ligação entre o PC e a rede de autómatos

### 3.1.2.1 Comunicação utilizando o *Facon communication protocol*

O *OPC Server Fatek Facon* permite a comunicação entre o PLC e o computador usando o *Facon communication protocol*. Este servidor consegue controlar e supervisionar múltiplos PLC's. O *OPC Server Fatek Facon* utiliza diferentes vias de comunicação com o computador, tais como uma comunicação série ou recorrendo ao protocolo *Tcp/Ip*.

É utilizado o *ActiveX* fornecido pela *Fatek Automation Corp* para estabelecer a comunicação com o PLC recorrendo à porta de comunicação *Ethernet*. Para que a comunicação seja possível é necessário criar um projecto no *Facon Server*, dentro do qual se definiu os parâmetros da comunicação:

- Device: Tipo de comunicação (RS232/Modem/Internet) e localização do PLC a utilizar
- Station Number: n.º associado à estação a utilizar
- Group: cada grupo criado tem a si associado determinadas variáveis para controlar.
- Item: cada item corresponde a uma variável, sendo possível definir o seu tipo (decimal, binário, hexadecimal ou *string*)

A Figura 36 apresenta um exemplo de um projecto criado no *Facon Server*.

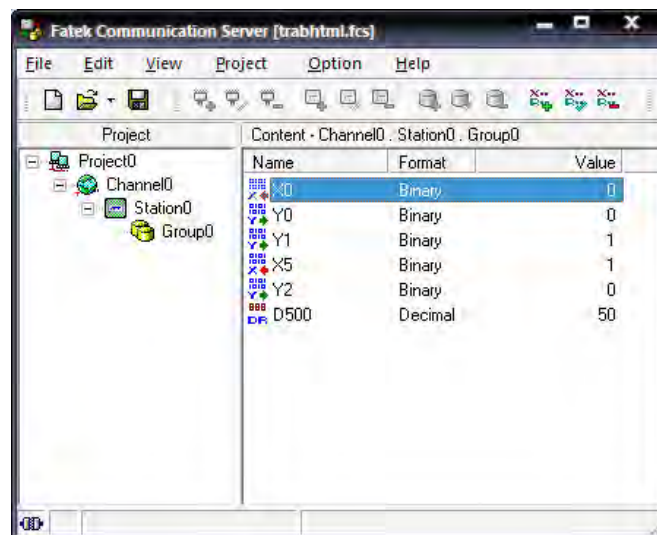


Figura 36 - Exemplo de um projecto criado no *Facon Server*.

Usando esta forma de controlo e monitorização da rede de autómatos por parte do PC, é possível recorrer a um conjunto de funções que efectuem de uma forma simples o controlo e a leitura de variáveis dos autómatos. Exemplos desse conjunto de funções são as seguintes:

- *Openproject()*: esta função abre o projecto criado anteriormente, para que funcione correctamente é necessário colocar como argumento a localização do ficheiro;
- *Connect()*: esta função estabelece a conexão com o PLC;
- *Disconnect()*: esta função interrompe a comunicação com o PLC;
- *Setitem()*: esta função escreve na variável o valor desejado, recebendo como argumentos uma *string* com a localização da variável no projecto (ex: "Channel0.Station0.Group0"), a variável (ex: "Y0") e o valor (ex: 0);
- *Getitem()*: esta função lê o valor de uma determinada variável, recebendo como argumentos uma *string* com a localização da variável no projecto (ex: "Channel0.Station0.Group0") e a variável (ex: "Y0").

### 3.1.2.2 Comunicação utilizando o protocolo de mensagens proposto

A comunicação entre o PC e a rede de autómatos pode ser feita através de ligação RS232 ou *Ethernet* através do protocolo de mensagens proposto anteriormente. A Figura 37 representa a estrutura da mensagem proposta.

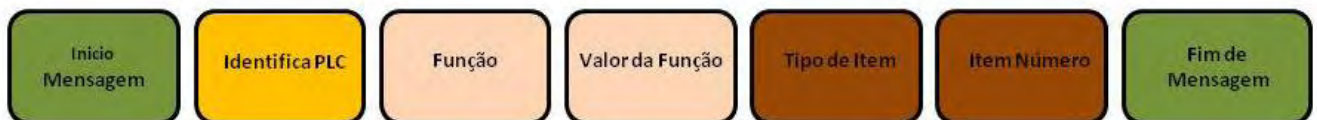


Figura 37 - Estrutura da Mensagem Proposta

A diferença para o método anterior reside no facto de, em vez de utilizar o *Facon communication protocol* para escrever nas posições de R+0 até R+6, a aplicação em Visual Basic desenvolvida é capaz de criar a mensagem adequada e de interpretar a resposta quando pede a leitura de uma variável de determinado autómato. A interpretação por parte do autómato



da mensagem recebida pelo PC realiza-se de forma idêntica à aplicada ao processo de comunicação entre autómatos.

### 3.1.3 Comunicação X10 – PC

A integração de sistemas de domótica na arquitectura proposta é da responsabilidade do PC. A integração com o sistema de domótica X10 é garantida pela comunicação entre o módulo CM11, representado na Figura 39, e o PC. Esta comunicação é estabelecida sobre uma ligação RS 232.

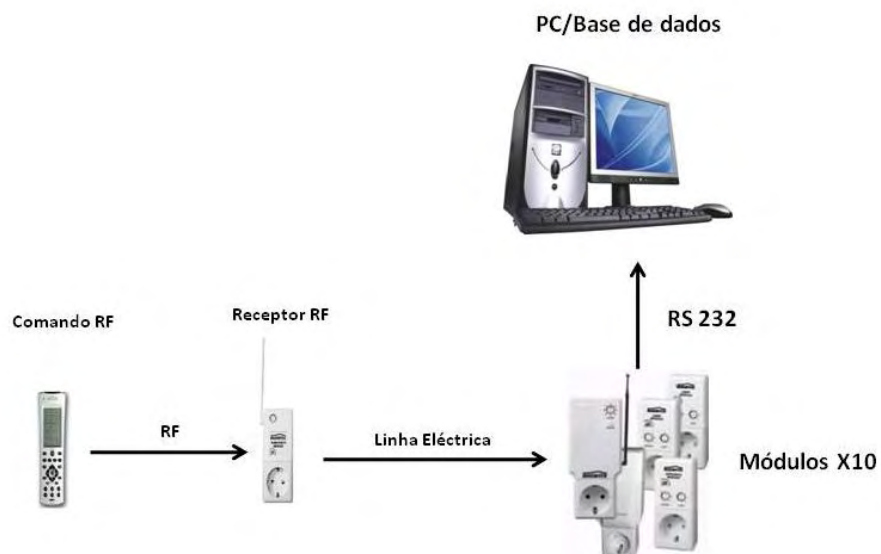


Figura 38 - Esquema de ligação entre o sistema de domótica X10, um comando RF, e o PC.



**Figura 39- Módulo CM 11 com saída *USB* que permite a ligação entre o PC e o sistema de domótica X10.**

Foi desenvolvida uma aplicação em Visual Basic que permite a interpretação dos sinais X10 que circulam na rede eléctrica. O módulo CM 11 apercebe-se do sinal e reencaminha determinados caracteres, que a aplicação desenvolvida interpreta, reencaminhando os dados para a base de dados.

#### ***3.1.3.1 Comunicação RF-CM11-PC***

Como prova da integração e flexibilidade da arquitectura proposta, desenvolveu-se a possibilidade de controlar um equipamento de Qbus ou uma saída de um autómato através do comando rádio frequência do sistema de domótica X10. O comando de rádio frequência emite o sinal correspondente para o receptor de X10, que volta a emitir o sinal correspondente na linha eléctrica, sendo assim captado pelo módulo CM 11. Em seguida este módulo envia os caracteres correspondentes por RS 232 para aplicação em Visual Basic, que está preparada para associar estes caracteres ao módulo de Qbus ou saída associada a um autómato, actualizando a informação devida na tabela de SQL. A Figura 40 esquematiza o processo descrito.

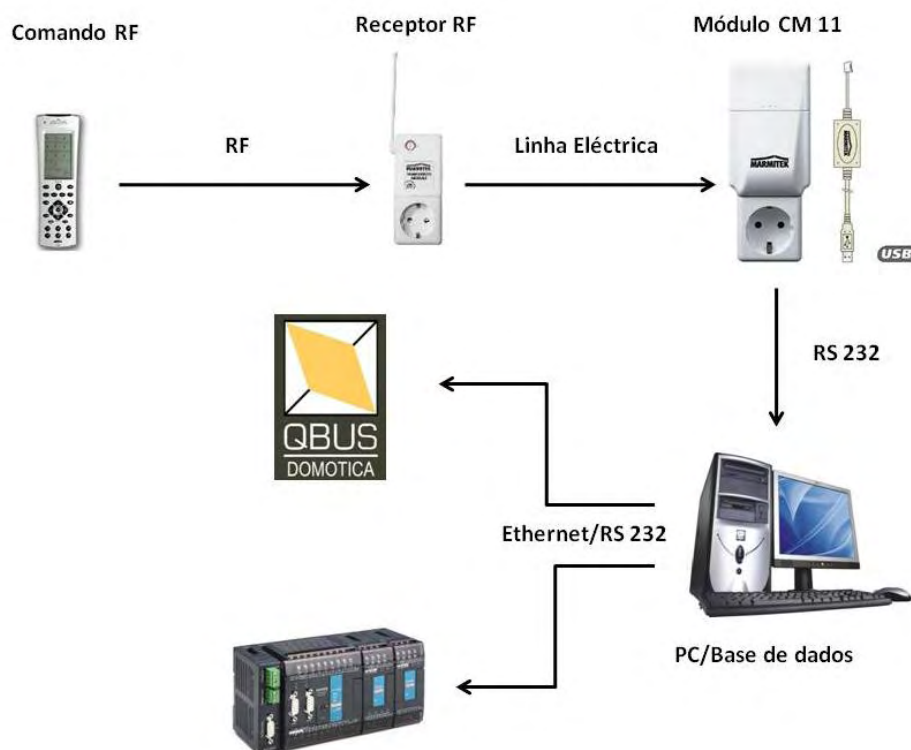


Figura 40- Esquema de comunicação entre uma ordem dada por RF para o controlo de um equipamento Qbus ou de um equipamento ligado a um autómato.

### 3.1.4 Comunicação Qbus - PC

Na arquitectura proposta a comunicação entre o quadro Qbus e o PC é feita através do protocolo Tcp/Ip, como se pode observar na Figura 41. O software desenvolvido em Visual Basic integra no seu projecto o conjunto de funções da biblioteca *qbuscom.dll* disponibilizado pelo próprio fabricante deste protocolo.

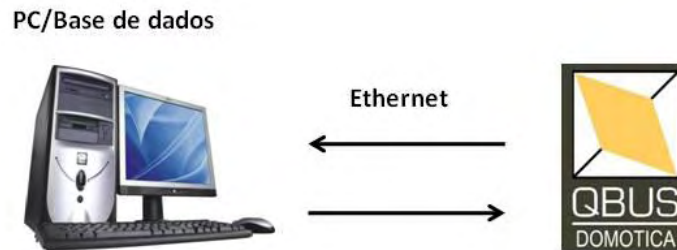


Figura 41 – Esquema da comunicação PC – Qbus.

A biblioteca `qbuscom.dll` oferece um conjunto de funções para diversas linguagens. Ao adicionar-se esta biblioteca de funções ao projecto no Visual Basic obtêm-se um conjunto diversificado de funções que permite controlar e monitorizar um quadro Qbus. Assim, as funções que aqui são descritas correspondem às funções mais utilizadas, havendo um conjunto de outras que poderiam ter sido utilizadas em função do que se pretendia controlar ou monitorizar no quadro Qbus.

A função que se utiliza para estabelecer uma ligação Tcp/Ip com o quadro Qbus é a função: `Qbus.OpenTCP("Ip Controlador", Porta, "Password")` (1). Se o valor que retorna depois da função executada for "0", indica que a ligação foi bem sucedida, caso contrário não.

A função que se utiliza para interromper uma ligação com o quadro Qbus é a função: `Qbus.CloseTCP()` (2). Se o valor que retorna depois da função executada for "0", indica que a comunicação foi bem sucedida, caso contrário não.

Outra das funções usadas na aplicação desenvolvida em Visual Basic, é a função que permite escrever ou ler variáveis. Esta função, da forma: `Qbus.Ctl_Online(["Readorwrite"], [address], [Mode], [data], [Number], [Offset])`, apresenta diversos campos, que têm de ser previamente confirmados. O campo ["Readorwrite"] define se a função é de leitura, ["R"], ou de escrita ["W"], o campo [address] define o endereço que vai ser objecto de escrita ou de leitura e o campo [Mode] define o tipo de endereço que se pretende ler ou escrever.

### 3.1.5 Comunicação PC – Base de dados

A arquitectura proposta pressupõe a utilização de uma base de dados que tem a função de guardar os parâmetros do sistema de domótica. Esta base de dados é actualizada pela aplicação desenvolvida em Visual Basic. A Figura 42 representa esquematicamente a ligação entre a aplicação Visual Basic e a base de dados.

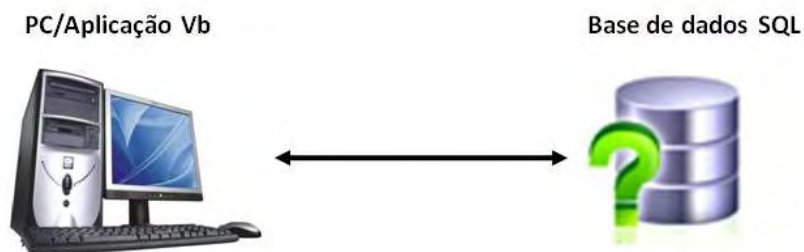


Figura 42- Representação esquemática da ligação entre o Pc/Aplicação Vb e a base de dados.

O servidor MySQL foi o escolhido para guardar o estado dos sensores e actuadores da habitação. Este servidor de base de dados foi desenvolvido originalmente para lidar com bases de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta exigência profissional por diversos anos com resultados satisfatórios.

A opção pelo servidor MySQL deve-se em parte ao facto de este interagir facilmente com o *Microsoft Visual Studio* e com a linguagem PHP. O servidor MySQL apresenta também um conjunto adicional de características que justificam a sua escolha:

- é pouco exigente quanto a recursos de hardware;
- é um software livre;
- tem um excelente desempenho e estabilidade;
- é compatível com o Microsoft Visual Studio e com a linguagem PHP usada na pagina *web* desenvolvida;
- portabilidade (suporta praticamente qualquer plataforma actual) ;
- é de fácil utilização.

Tendo em vista a flexibilidade da estrutura proposta deve-se construir a tabela de domótica do sistema, ou seja, definir o número de entradas em função da saída que se pretende activar.

Designação do actuador	Nº Entradas	Entrada 1	Estado Entrada 1	Operador Lógico 1	Entrada N	Estado Entrada N	Endereço Actuador	Valor Actuador
Relé Qbus	2	A2X1	OFF	AND	A1X0	OFF	QB15	OFF

**Figura 43 - Campos da tabela SQL que controla e monitoriza o sistema de domótico proposto.**

O campo “designação” serve simplesmente para atribuir um nome à saída que se pretende controlar ou monitorizar. O campo “número de entradas” define o número de entradas que define o estado da saída. O campo “Entrada 1” define o endereço da entrada 1 na arquitectura proposta. O campo “estado da entrada 1” tal como o próprio nome indica, representa o estado da entrada 1.

O campo “operador lógico 1” indica qual o operador lógico entre as entradas  $n$  e  $n + 1$  de modo a definir se o estado da saída deve ser alterado ou não.

O campo “endereço do actuador” indica qual a variável do sistema de domótica que se pretende controlar ou monitorizar e o campo “valor do actuador” indica estado do actuador.

Foi criado um sistema de endereçamento para as entradas e saídas do sistema protótipo proposto, que vai ser descrito nos próximos capítulos.

### ***3.1.5.1 Endereçamento das saídas e das Entradas da arquitectura proposta***

O facto de a arquitectura proposta suportar diversas entradas e saídas de sistemas de domótica diferentes, conduziu à necessidade de criar um sistema de endereçamento que permitisse uma diferenciação clara entre as tecnologias pertencentes à arquitectura proposta que inclui os autómatos, X10 e o Qbus. Assim utilizou-se as letras A, X e Q como primeira forma de distinção nos endereços.

### **3.1.5.1.1 Endereçamentos nos Autómatos**

Nos autómatos houve necessidade de criar um campo no endereço que define a estação do PLC. Assim, o segundo campo do endereçamento de um autómato corresponde ao número da estação onde se encontra a variável que se pretende controlar ou monitorizar.

No endereçamento de uma variável de um autómato existe ainda a necessidade de diferenciar se é uma entrada digital, saída digital, registo, registo de data ou posição de memória. Com este propósito foi definida a seguinte nomenclatura:

X – entradas Digitais;

Y – saídas Digitais;

R – registos;

D – registos de Data;

M – posições de Memória;

Quando se trata de variáveis dos autómatos o endereço termina com o número do endereço físico da variável. Por exemplo: A2X3 corresponde à entrada digital 3 do autómato número 2.

### **3.1.5.1.2 Endereçamento no Qbus**

O endereçamento do Qbus é definido no primeiro campo pela letra Q, seguido pelo endereço do equipamento que se pretende controlar. Este equipamento pode ter vários modos de funcionamento. Este campo ocupa sempre dois caracteres uma vez que todos os modos de operação dos vários componentes Qbus são definidos por dois campos. O campo seguinte corresponde ao endereço físico do módulo Qbus que se pretende controlar ou monitorizar. Por exemplo, o código QBI5 indica que se pretende ler o relé de Qbus "BI", com o endereço "5".

#### **3.1.5.1.3 Endereçamento no X10**

O próprio protocolo X10 implementa um sistema simples de endereçamento que utiliza 16 códigos do aparelho (1-16) e 16 códigos de casa (A-P). Assim, houve a necessidade de o integrar na tabela da base de dados, diferenciando-os claramente dos outros. Definiu-se que o primeiro campo do endereço é um X e antecede o código da casa. Este par de caracteres é seguido de outro carácter com o código do aparelho. Por exemplo, o endereço XA2, indica que se pretende actuar um equipamento da rede X10, “X”, definido pelo endereço “A2”.



## 4 Implementação e análise do desempenho

Um dos objectivos deste trabalho é implementar uma solução integrada. A arquitectura proposta anteriormente integra três tecnologias diferentes. Os critérios para a escolha das tecnologias para o sistema protótipo desenvolvido tiveram por base diferentes parâmetros que irão ser descritos nos parágrafos seguintes.

A integração do QBUS no sistema protótipo desenvolvido justifica-se pelo facto de apresentar serviços muito semelhantes ao protocolo EIB, sendo por isso uma das tecnologias mais utilizadas no mercado. Por esse motivo faz todo o sentido que o Qbus faça parte do protótipo a implementar. Outro factor decisivo deve-se ao facto de o site do Qbus disponibilizar uma completa biblioteca de funções para diversas linguagens de programação nomeadamente para o *Visual Basic*.

O protocolo X10 é outra das tecnologias presentes no protótipo. A sua inclusão é facilmente justificável, uma vez que esta tecnologia permite que a arquitectura proposta tenha uma maior flexibilidade. De facto, o X10 é uma solução para habitações antigas ou já construídas e sem possibilidade de alterar a cablagem convencional aplicada. Outra justificação encontrada para a inclusão desta tecnologia é a excelente relação custo/benefício e a facilidade de instalação e de operação.

A integração de uma rede de autómatos no sistema protótipo assume principal relevância dado que a arquitectura proposta se baseia na utilização de autómatos para o controlo da habitação. A arquitectura proposta permite também a sua integração com sistemas já existentes como o caso do X10 e do Qbus.

## 4.1 Estrutura do sistema protótipo proposto

O protótipo implementado tem uma componente de *hardware* em que foram escolhidos equipamentos disponíveis no mercado segundo os critérios explicados anteriormente. Outra componente do sistema protótipo consistiu no desenvolvimento de *software* que permite o controlo e a monitorização dos diversos equipamentos.

### 4.1.1 Hardware utilizado

#### 4.1.1.1 Hardware X10

O X10 é uma das tecnologias integradas no sistema protótipo proposto. Os módulos X10, que fazem parte do sistema, foram escolhidos em função do que é mais comum controlar numa casa inteligente.

##### Módulo de Aparelho de tomada

Este Módulo X10, Figura 44, permite controlar aparelhos ligados a tomadas de parede. Com este módulo podemos, por exemplo, controlar remotamente aquecedores, candeeiros com lâmpadas fluorescentes ou outros aparelhos.



Figura 44 - o módulo de aparelho X10 [13].

### Interruptor de Calha DIN

Interruptor de Calha DIN para cargas de até 3600W, Figura 45, pode ser controlado remotamente através de comandos X10, ou através de interruptores convencionais, de pressão ou temporizados. O seu estado pode ser verificado pelo indicador luminoso existente. Com este interruptor é possível controlar aparelhos e luzes utilizando o sistema X10. Possui terminais para adicionar botões de pressão extra, o que significa que o interruptor funciona em comutação de escada. O interruptor tem um selector que pode ser programado para “Sempre On”, “Sempre Off” ou “Controlo Remoto” (auto). Este interruptor responde a comandos X10 On, Off e All Off (“All Appliances Off” e “All Lights On”)[12].



**Figura 45 - interruptor de Calha DIN [13].**

### Receptor radiofrequência - Módulo Transmissor X10

Este módulo implementa a função de conversão de sinais e radiofrequência dos comandos X10 em sinais X10 na rede eléctrica (Figura 46). Este módulo é necessário para controlar os receptores X10 a partir de qualquer comando RF X10 [12].



**Figura 46 - receptor de radiofrequência de sinais X10 [13].**

### Módulo CM 11

O Módulo CM 11 (Figura 47), permite o envio e a recepção de mensagens entre o PC e a rede eléctrica. Este módulo capta os sinais eléctricos e transforma-os em mensagens, que envia para o computador através da porta RS 232. Recebe também mensagens oriundas do PC e transforma-as nos sinais eléctricos correspondentes.



**Figura 47- Módulo CM 11[13].**

### Comando de Radiofrequência

O comando de radiofrequência (Figura 48) presente no sistema implementado tem a capacidade de enviar comandos X10 por radiofrequência e permite o controlo de televisões, vídeos ou outros componentes audiovisuais.



**Figura 48 - Comando de radiofrequência [13].**

#### 4.1.1.2 Hardware de Qbus

O conjunto de módulos Qbus integrados no sistema protótipo, foram um módulo *Ethernet*, um interruptor que controla outros dois módulos, um relé e um módulo *dimmer*.

##### Interface de Ethernet - ETH02

Este módulo (Figura 49) permite a comunicação por Tcp/Ip entre o quadro Qbus e o PC ou com o *gateway* de *Ethernet* existente na habitação.



Figura 49 – Módulo ETH02, interface de *Ethernet* [14].

##### Módulo Relé - REL04

O Módulo relé permite a ligação para 4 equipamentos que podem ser controlados de forma independente. Este relé não tem associada nenhuma função específica, podendo associar-se a ele diversas funções (*On/Off*, temporizador, termóstato, etc).

##### Módulo Dimmer – Dim04-300

Este módulo (Figura 50) permite regular a intensidade luminosa de um circuito de iluminação integrado num sistema Qbus.



Figura 50 - Dim04-300 – Módulo *Dimmer* [14].

#### Interruptor – SWI04

O interruptor (Figura 51) permite o controlo manual do *dimmer* e do relé integrado no sistema protótipo.



Figura 51 - SWI04 – Interruptor Qbus [14].

#### **4.1.1.3 Rede de Autómatos**

Os autómatos utilizados no protótipo implementado foram os autómatos da *Fatek Automation Corp.* Foram utilizados dois modelos de autómatos diferentes, o modelo *FBs-60MC* com módulo de comunicações *Fbs-CM25E* e o modelo *FBs-20MC* com o módulo de comunicações *FBs-CM55E*, simulando desta forma uma rede de autómatos, entre os quais foi implementado o protocolo de comunicação apresentado no capítulo anterior.

O modelo *FBs-20MC*, representado na Figura 53, basicamente é caracterizado por ter doze entradas digitais, oito saídas digitais, uma porta *Ethernet*, uma porta *USB*, uma porta *RS 485*, duas entradas e uma saída analógica [15].

O modelo *Fbs-60MC*, representado na Figura 52, possui trinta e seis entradas digitais, vinte e quatro saídas digitais. Integra ainda por ter uma porta *RS232*, uma porta *USB*, uma porta *Ethernet* e uma porta *RS 485* [15].



Figura 52 – Autómatos Fatek Corp, modelo Fbs-60MC [15].



**Figura 53– Autómato Fatek Corp, modelo Fbs-20MC [15].**

## **4.1.2 Software desenvolvido**

Para a implementação do sistema protótipo foi desenvolvido uma aplicação em *Microsoft Visual Studio 2008* usando a linguagem *Visual Basic* produzida pela empresa *Microsoft*. Esta linguagem é controlada por eventos e é caracterizado por possuir um ambiente de desenvolvimento integrado, gráfico, facilitando assim a construção da interface das aplicações. Este factor foi decisivo na opção por esta linguagem de programação para o desenvolvimento do protótipo.

Com recurso a esta linguagem de programação foram desenvolvidas um conjunto de funções que permitem a integração das diversas tecnologias num único projecto.

### **4.1.2.1 Software X10**

Foi desenvolvida uma aplicação que permite ao computador comunicar com os equipamentos da rede X10. Este programa utiliza a interface RS232 do computador para comunicar com o modem CM11, que por sua vez faz a interface com a rede X10.

Assim, foi desenvolvido um conjunto de funções como a abertura e o fecho da porta de comunicações com o sistema X10. Para tal foi necessário definir os parâmetros de comunicação entre o PC e o módulo CM 11:

- Taxa de transmissão: 4800 bps
- Paridade: nenhuma
- Nº de bits de dados: 8
- Nº de stop bits: 1

Foram desenvolvidas outras funções que permitem o controlo dos módulos de X10, permitindo enviar um conjunto de operações (*On/Off, Dim, all units on/off* ). Foi ainda desenvolvido um

conjunto de funções que permite associar saídas digitais, ou módulos de Qbus, a sinais X10 emitidos por radiofrequência pelo comando rádiofrequência presente no sistema de domótica protótipo. Tal foi conseguido através da exploração da propriedade *oncomm* do objecto *mscomm*. No anexo 1 é apresentado o protocolo de comunicação X10 entre o PC e o módulo de interface X10 com a linha eléctrica, que se relevou bastante útil ao processo de desenvolvimento de *software*.

#### ***4.2.2.2 Software Qbus***

A integração do Qbus na aplicação desenvolvida é conseguida através da utilização da biblioteca de funções *qbuscomm.dll* fornecida pela Qbus. Foi também desenvolvido um conjunto de funções que permite controlar e monitorizar o conjunto de módulos Qbus presentes no protótipo.

#### ***4.1.2.3 Software desenvolvido para a rede de Autómatos***

A rede de autómatos presentes no sistema protótipo é gerida na aplicação desenvolvida através do *OPC Server Fatek Facon* que permite a comunicação entre o PC e o autómato através do *Facon communication protocol*. O autómato que está ligado ao PC contém o estado das variáveis dos restantes autómatos da rede de autómatos, que comunicam entre eles seguindo o protocolo de mensagens proposto no capítulo anterior. A aplicação *Visual Basic* monitoriza o sistema através do *Facon communication protocol*, efectuando uma leitura da variável correspondente no autómato a que se encontra ligado. Se o autómato pretende controlar uma variável da rede de autómatos, a aplicação gera uma mensagem que é enviada para o autómato que está ligado ao PC e este redirecciona-a para a rede de autómatos, que interpreta a mensagem e efectua a escrita da variável pretendida no autómato para o qual a mensagem é direccionada.

#### ***4.1.2.4 Funcionamento base do Software desenvolvido***

No decorrer deste trabalho foi desenvolvido um conjunto de funções que permite o funcionamento pretendido do modelo protótipo implementado. O objectivo dos próximos parágrafos é descrever de uma forma sucinta as principais funções e eventos gerados pela aplicação desenvolvida.



- **Evento *Form Load***

Este evento é associado ao executar do programa. Neste evento é estabelecida a comunicação com os sistemas de domótica (X10, Qbus e rede de autómatos) e com a base de dados, sendo ainda iniciada a interface gráfica. Para tal, é necessário efectuar uma leitura do estado dos equipamentos, com o objectivo de actualizar a interface gráfica logo após o executar do programa. Durante este evento é iniciado o temporizador geral do sistema.

- **Temporizador geral**

O Temporizador geral assume especial relevo na aplicação desenvolvida uma vez que é o responsável pela actualização da interface gráfica e pela constante monitorização das entradas e saídas dos equipamentos. A actualização da interface gráfica é efectuada em todos os ciclos deste temporizador através da análise de um *array* com o estado dos equipamentos. A monitorização dos equipamentos que definem a actuação de outros é efectuada pela função *Le\_Entradas*. A actuação de todos os equipamentos é realizada através da função *Escreve\_Saídas*.

Em cada ciclo deste temporizador é efectuada a monitorização e o controlo de um equipamento.

- **Temporizadores secundários**

Existem dois temporizadores secundários na aplicação desenvolvida. É atribuída a nomenclatura de temporizadores secundários porque permitem o funcionamento de mecanismos complementares do protótipo desenvolvido. Um deles é responsável por gerir eventuais eventos relacionado com a temporização do funcionamento dos equipamentos. Na aplicação desenvolvida existe outro temporizador que gere eventos relacionados com a simulação aleatória de presença. Os eventos relacionados com os temporizadores secundários estão activos quando o utilizador activa as funções relacionadas.

- **Função *Le\_Entradas***

Esta função tem o objectivo de monitorizar os equipamentos. Esta função tem um argumento de entrada, o número da linha da tabela de domótica desenvolvida. Esta começa por analisar o número de entradas definidas na base de dados. Depois de interpretar o número de entradas, vai proceder à sua análise. Em função do estado das entradas, do número de entradas e do operador lógico como se relacionam as entradas, é devolvido um valor. Este pode assumir valores diferentes:

1 – Significa que a saída correspondente deve ser ligada;

0 – Significa que a saída correspondente deve ser desligada;

77 – Significa que ocorreu um erro.

O número de entradas para activar um equipamento é 0 quando este se encontra a ser controlado remotamente.

O valor devolvido por esta função é guardado num vector, com o intuito de guardar o histórico sobre o último valor devolvido por esta função para o equipamento em causa.

- **Função *Escreve\_Saídas***

A função *Escreve\_Saídas* é responsável pela actuação dos equipamentos. Esta função tem três argumentos de entrada: o número da linha da tabela, o estado lógico (definido pela função *Le\_Entradas*) e o estado anterior do equipamento.

Esta função analisa primeiro os argumentos de entrada, verificando se há ou não uma ordem diferente da anterior para a actuação do equipamento definido pelo número da linha da tabela. Se existir então uma ordem diferente da anterior, esta função analisa o tipo de equipamento que pretende actuar e a função a implementar definido no endereço da tabela SQL desenvolvida. Por último, procede ao envio da ordem para o equipamento em causa.

Esta função retorna o estado do equipamento.

- **Função *Timmer\_on***

A função *Timmer\_on* é responsável pelo controlo dos equipamentos através de intervalos de tempo de funcionamento. Esta função tem três argumentos de entrada, uma *string* que indica quando activar o equipamento, uma *string* que indica quando desactivar o equipamento e número da linha da tabela desenvolvida. Esta função é executada quando o campo da tabela que define se existe uma temporização ou não, se encontra no estado activo. O funcionamento básico desta função é a comparação da hora do PC com as temporizações definidas nos argumentos de entrada. Quando a hora do PC é igual à hora definida no argumento que indica a activação do equipamento, esta função altera o modo como o equipamento é controlado para o modo de controlo remoto, e mantém-se neste estado até enviar a ordem de desactivação do equipamento. Quando é enviada a ordem de desactivação do equipamento, o modo como o equipamento é controlado volta ao modo definido anteriormente pelo utilizador.

#### 4.1.2.5 Fluxogramas que descrevem o funcionamento do Software desenvolvido

Nesta secção são apresentados fluxogramas que esquematizam o funcionamento do software desenvolvido em Visual Basic. Na Figura 54 é apresentado um fluxograma que explica o funcionamento da aplicação visual Basic que controla e monitoriza sistema.

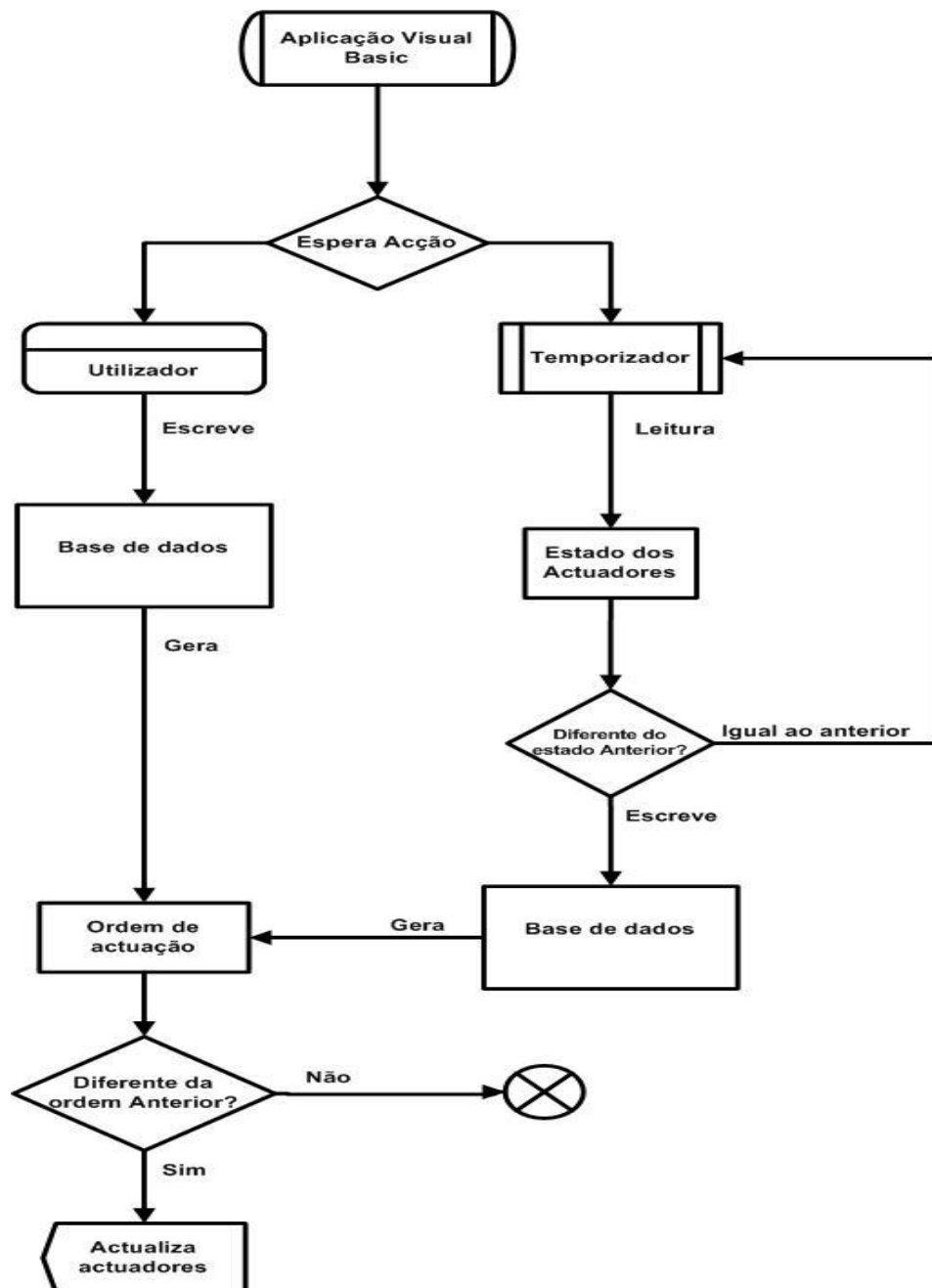


Figura 54 - Funcionamento do software quando o utilizador controla ou monitoriza o sistema.

Na Figura 55 é apresentado um fluxograma que mostra como o utilizador pode interagir com o sistema.

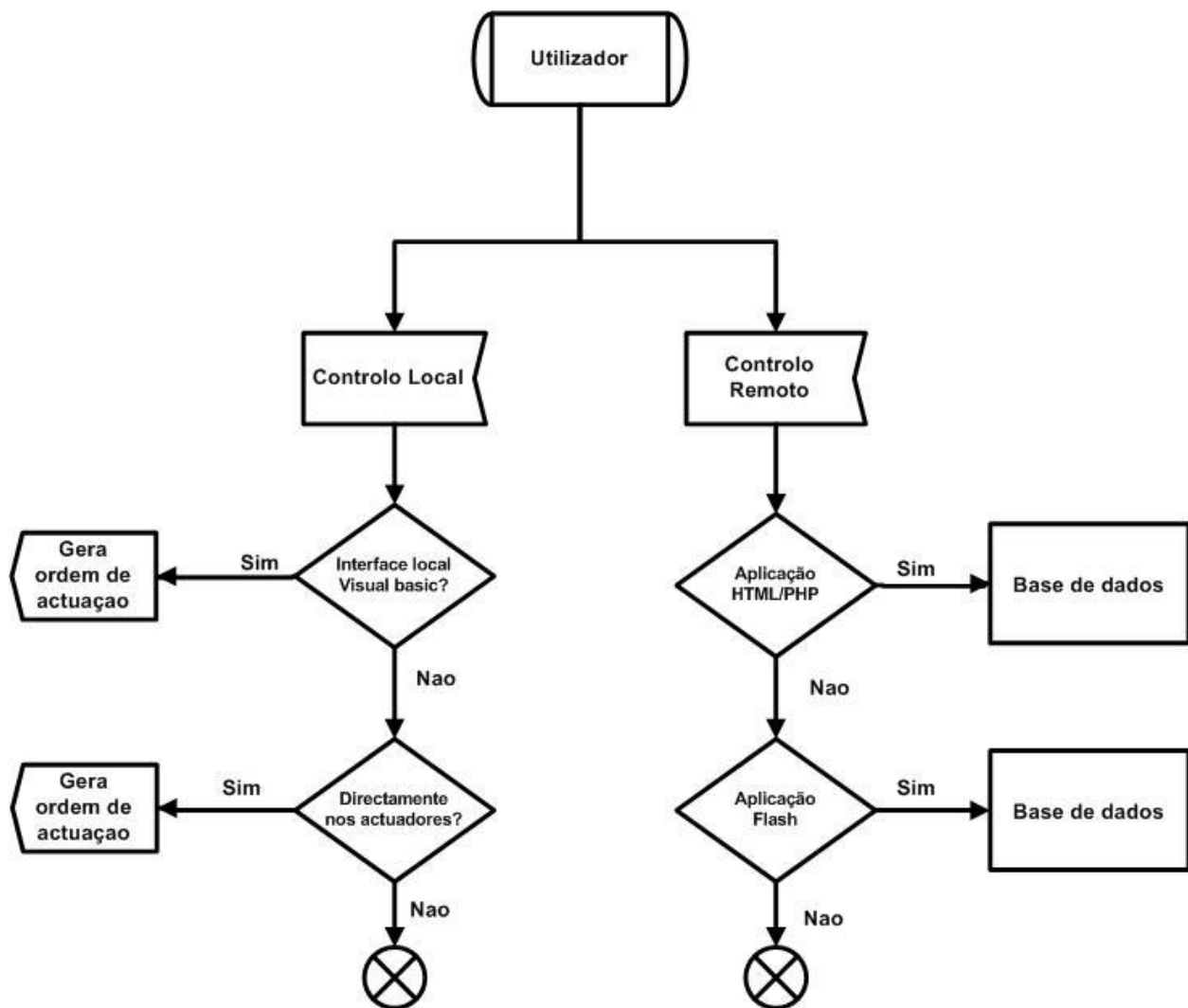


Figura 55 - Esquema que representa o funcionamento do software com controlo por parte do utilizador.

#### 4.1.2.6 Interfaces gráficas realizadas

As interfaces gráficas assumem um papel importante porque é a forma como o utilizador vai poder controlar e monitorizar os mecanismos pertencentes ao sistema domótico. Parte dos objectivos do trabalho é desenvolver uma interface de utilização simples e amigável e uma interface que tivesse a capacidade de acesso e gestão remota. Neste contexto foi desenvolvida uma interface gráfica em Visual Basic, e uma interface em HTML e PHP que tentam corresponder a estes objectivos.

##### 4.1.2.6.1 Interface Gráfica – Controlo e monitorização local

A linguagem Visual Basic integrada no *Microsoft Visual Studio* é a linguagem de programação escolhida, porque, para além do estabelecimento de funções que comandam os diversificados equipamentos, permite desenvolver interfaces gráficas de uma forma elementar e bastante amigável do ponto de vista gráfico. Esta linguagem de programação baseia-se na gestão de eventos onde cada evento está associado a uma mudança na base de dados desenvolvida para este protótipo. A aplicação em *Visual Basic* regista a mudança e age em conformidade, comandando o protótipo. Assim, a Figura 56, ilustra a interface gráfica desenvolvida.



Figura 56 – Panorama Geral da Interface Gráfica desenvolvida.

De uma forma geral, está representado a vermelho o menu de funções básicas do sistema, a amarelo pode monitorizar-se os diversos mecanismos existentes no modelo protótipo, a verde temos o menu das funções inteligentes desenvolvidas, a azul pode observa-se como está a ser efectuado o controlo dos diversos mecanismos pertencentes ao protótipo, a cor de laranja pode observa-se o estado das comunicações entre o servidor e os sistemas de domótica presentes no protótipo.

O menu de funções básicas (Figura 57) permite ao utilizador ter um controlo geral sobre determinados sistemas funcionais da sua habitação. O item “Modo Automático” tem a função de atribuir ao sistema o controlo automático dos equipamentos do sistema protótipo proposto. O controlo automático é definido na configuração do sistema por parte do utilizador.



Figura 57 – Menu de funções básicas.

No menu de funções inteligentes (Figura 58) foram definidas determinadas funções e cenários.



**Figura 58 – Menu inteligente.**

O modo cinema é caracterizado por reduzir a intensidade luminosa da sala, fechar os estores e actuar sobre os componentes audiovisuais.

O modo despertar, quando activo através de uma temporização definida ou de um controlo manual, é caracterizado pelo ligar do termoacumulador e pela activação do controlo geral da temperatura.

O modo anoitecer, quando activo através de uma temporização definida ou de um controlo manual, é caracterizado pelo fecho dos estores e pela activação do controlo geral da temperatura.

O modo simulação de presença, quando activo através de uma temporização definida, é caracterizado pela activação de um módulo de iluminação aleatório em intervalos de tempo diferentes durante a temporização definida.

Na interface desenvolvida é possível visualizar o estado praticamente em tempo real (Figura 59) dos equipamentos pertencentes ao protótipo.





Figura 59 – Monitorização dinâmica do protótipo.

A segurança no modelo protótipo está associada só à detecção de intrusão. No protótipo proposto pode visualizar-se onde está a ser detectada presença não autorizada (Figura 60).

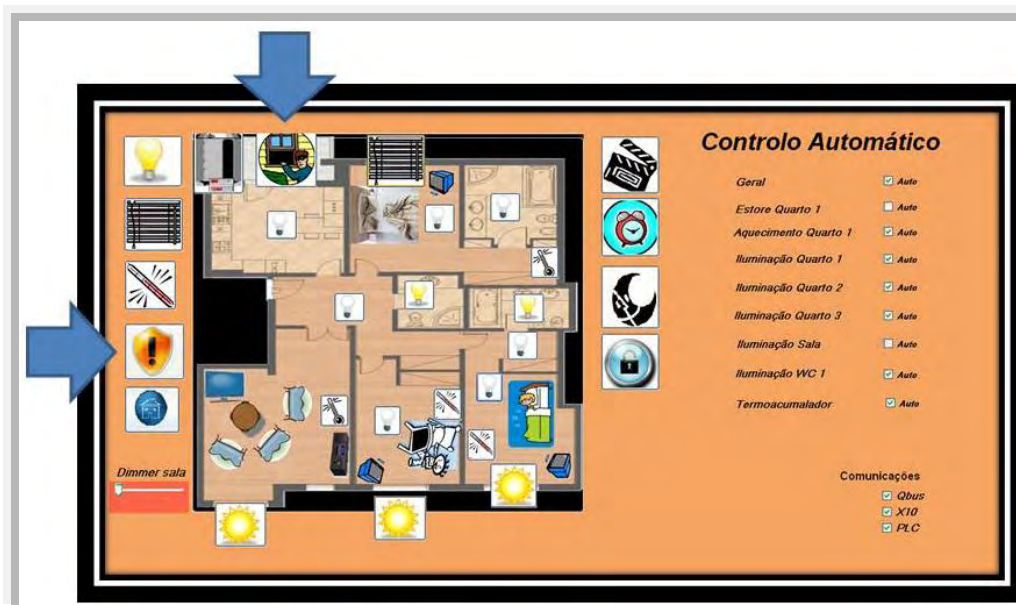


Figura 60 – Detecção de intrusão no lado “norte” da habitação.

A forma como os equipamentos são controlados é definida pelo utilizador. O módulo pode ser controlado de forma automática através de uma ou varias entradas do sistema (definido na tabela de domótica desenvolvida), ou de uma forma manual através desta interface ou através da Web. O utilizador pode determinar o tipo de controlo de cada equipamento, bastando para isso activar as caixas de selecção (Figura 61) se pretende modo automático ou desactivar se pretende controlo manual.

A Figura 61 mostra também o estado da comunicação entre o PC e os diversos sistemas de domótica pertencente ao sistema proposto.

**Controlo Automático**

<i>Geral</i>	<input type="checkbox"/> Auto
<i>Estore Quarto 1</i>	<input checked="" type="checkbox"/> Auto
<i>Aquecimento Quarto 1</i>	<input checked="" type="checkbox"/> Auto
<i>Iluminação Quarto 1</i>	<input type="checkbox"/> Auto
<i>Iluminação Quarto 2</i>	<input type="checkbox"/> Auto
<i>Iluminação Quarto 3</i>	<input checked="" type="checkbox"/> Auto
<i>Iluminação Sala</i>	<input type="checkbox"/> Auto
<i>Iluminação WC 1</i>	<input checked="" type="checkbox"/> Auto
<i>Termoacumulador</i>	<input checked="" type="checkbox"/> Auto

Comunicações

- ☒ Qbus
- ☒ X10
- ☒ PLC

**Figura 61 – Quadro de Comunicações e de Controlo**

No modelo protótipo foi desenvolvido um conjunto de funções que permitem aos equipamentos serem controlados através de temporizações. Estas podem ser definidas na interface básica. Para tal, é necessário definir as temporizações pretendida na caixa de texto. Em seguida, selecciona-se através do Menu de temporizações o equipamento para o qual estamos a definir a temporização, A imagem da Figura 62 ilustra o processo.

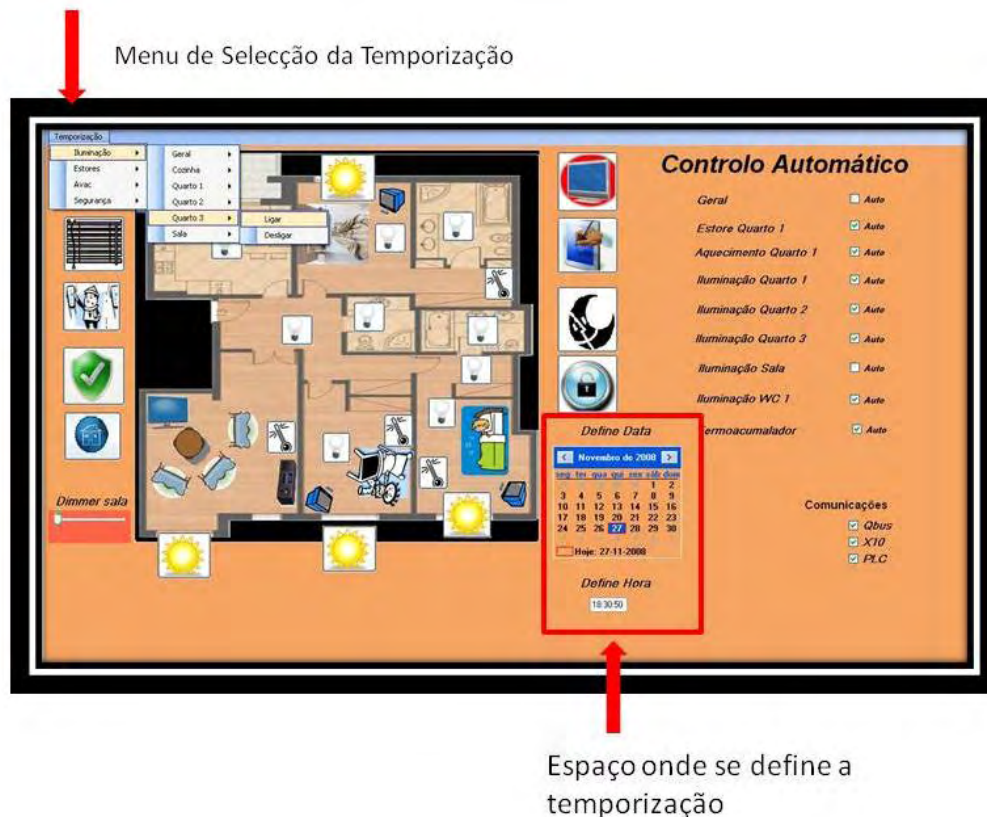


Figura 62 – Menu de temporizações.

A definição da temporização deve ser realizada seleccionando o dia e a hora pretendida para o controlo do equipamento pretendido.

#### 4.1.2.6.2 Interface Gráfica – Controlo e monitorização remota

A gestão e o controlo remoto em qualquer sistema domótico é nos dias que correm um serviço fundamental, uma vez que as tendências tecnológicas apontam para um uso generalizado da internet, nomeadamente da banda larga e das redes *wireless*.

#### 4.1.2.6.2.1 Pagina Web em HTML e PHP

Para o sistema protótipo proposto foi desenvolvido uma página *Web* em HTML e PHP, onde é possível efectuarmos o controlo e a monitorização do sistema protótipo proposto.

Optou-se pelas linguagens HTML e PHP porque linguagens permitem facilmente o acesso a uma base de dados MySQL.

De uma forma geral, o utilizador da página Web (Figura 63), está a escrever directamente na tabela de domótica que está no servidor. No servidor está também a aplicação em *Visual Basic* que interpreta estas alterações efectuadas remotamente.

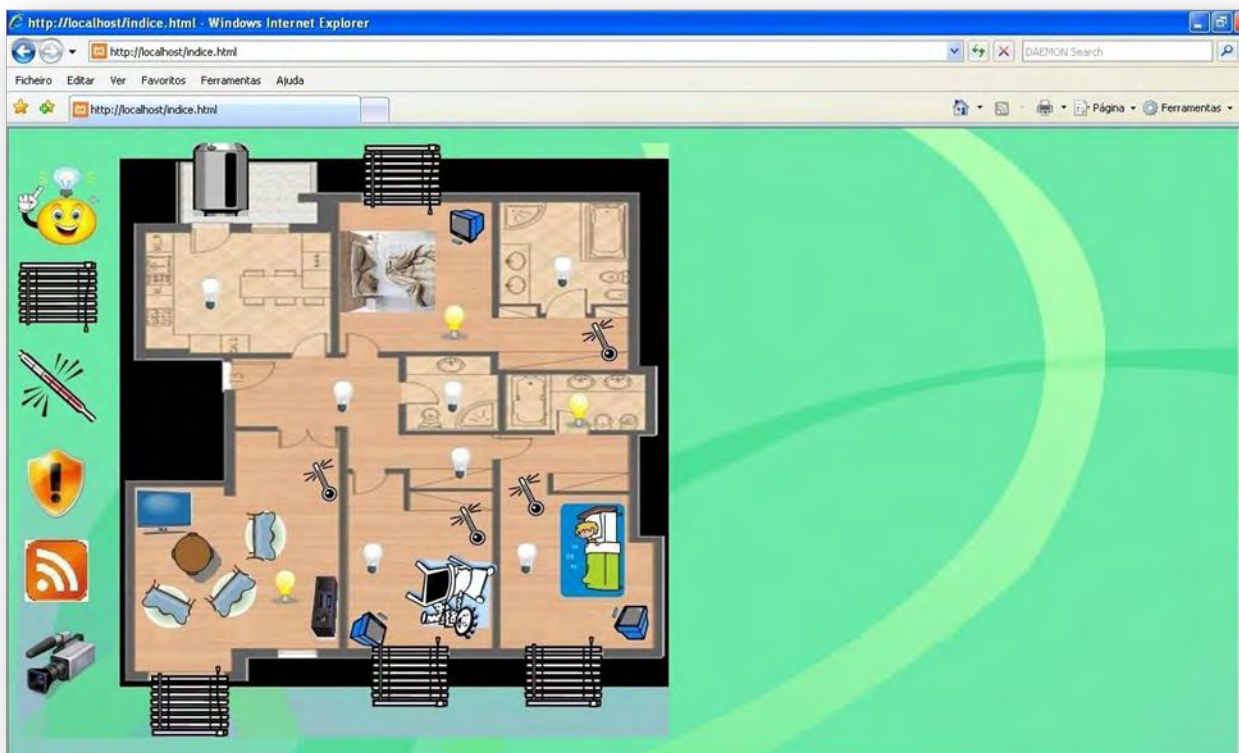


Figura 63 – Aspecto da página Web desenvolvida.



A Figura 64 ilustra a página Web desenvolvida usando delimitadores coloridos para identificar os principais componentes ou áreas de intervenção. O controlo remoto através da página Web desenvolvida é efectuada através de selecção do que se pretende controlar ou monitorizar no sistema protótipo proposto. Assim, a vermelho é representada a zona que serve de menu dos diversos sistemas funcionais presentes no protótipo desenvolvido. Desta forma, a cor azul representa a iluminação, a cor amarela o controlo dos estores e a cor laranja o controlo de temperatura e de humidade. O controlo geral de cada um dos sistemas funcionais pode ser visualizado ou controlado clicando no quadrado da cor correspondente que se encontra no menu (zona a vermelho). Por outro lado se quisermos um controlo ou a gestão apenas de um elemento do sistema protótipo clica-se no item correspondente.



**Figura 64 – zonas de controlo e gestão dos diversos módulos do protótipo.**

A segurança, tal qual na interface anterior, mereceu a devida importância no desenvolvimento da página Web (Figura 65). A página Web foi desenvolvida com o objectivo de se assemelhar

ao máximo em termos de funções e aspecto à interface anterior. Pode observar-se na lista do lado esquerdo a monitorização dos detectores de intrusão.



Figura 65 – Janelas desenvolvidas no protótipo que contemplam a segurança.

A nível gráfico, a Figura 66, mostra as diferenças entre a interface anterior e a interface Web desenvolvida. A verde tem-se o equivalente na interface anterior às funções inteligentes. Clicando neste item do menu pode visualizar-se a janela (Figura 67) que se assemelha em termos gráficos e de funções descritas anteriormente na interface desenvolvida em Visual Basic. O rectângulo de cor vermelha representa uma hiperligação para uma *webcam*, que simula a vídeo vigilância no protótipo. O rectângulo de cor amarela simboliza uma hiperligação para o regresso à janela representada na Figura 64.

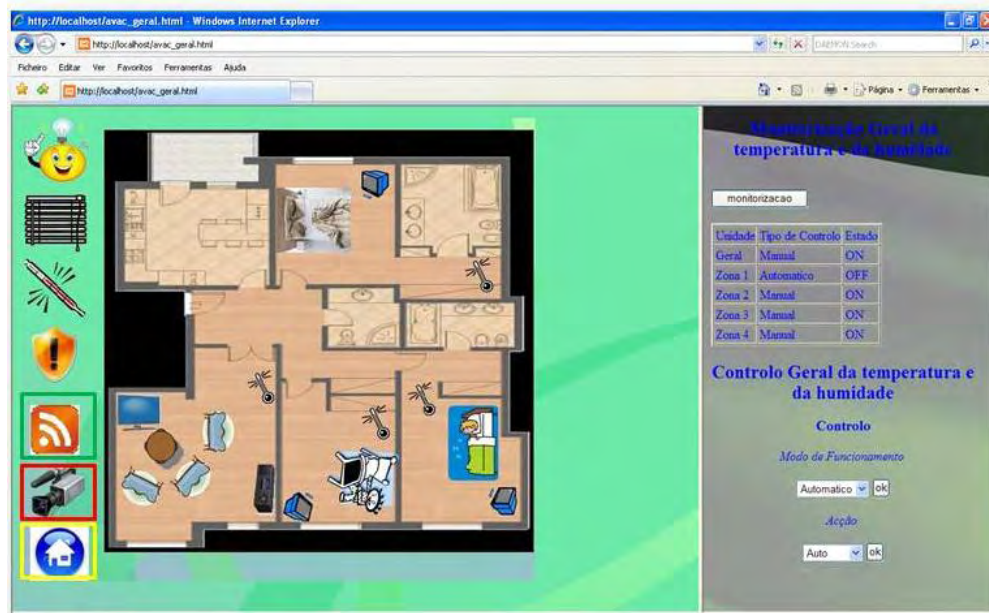


Figura 66 – Diferenças entre interface Web e Visual Basic.



Figura 67 – Menu de Funções “inteligentes”

#### 4.1.2.6.2.2 Pagina Web - Aplicação Flash

Para o sistema protótipo foi desenvolvida uma aplicação *flash* que permite o controlo e a monitorização dos equipamentos. O *Adobe Flash*, antes denominado de *Macromedia Flash*, é um *software* que permite a criação de animações interactivas que são integradas numa página Web. Habitualmente denomina-se *flash* os arquivos gerados pelo *Adobe Flash*, estes ficheiros têm a extensão “*.swf*” (*Shockwave Flash File*). A linguagem utilizada para gerar estes ficheiros é o *ActionScript*. A Adobe acaba de lançar a terceira versão desta linguagem, que é caracterizada por ter um numero maior de recursos que as versões anteriores e pela facilidade e rapidez de criação de aplicações para a Web.

A aplicação desenvolvida para efectuar o controlo e a monitorização dos equipamentos recorre aos ficheiros PHP desenvolvidos para a aplicação anterior. Assim, se o objectivo for efectuar o controlo de um determinado equipamento, a aplicação *flash* envia a mensagem correspondente ao ficheiro PHP, o ficheiro PHP recebe a mensagem e, em função do que lhe foi requerido pela aplicação *flash*, actualiza a tabela de SQL. A aplicação desenvolvida faz a continua monitorização do sistema através de um ficheiro PHP que, quando é solicitado, fornece à aplicação *flash* o estado dos equipamentos. A Figura 68 esquematiza este processo.

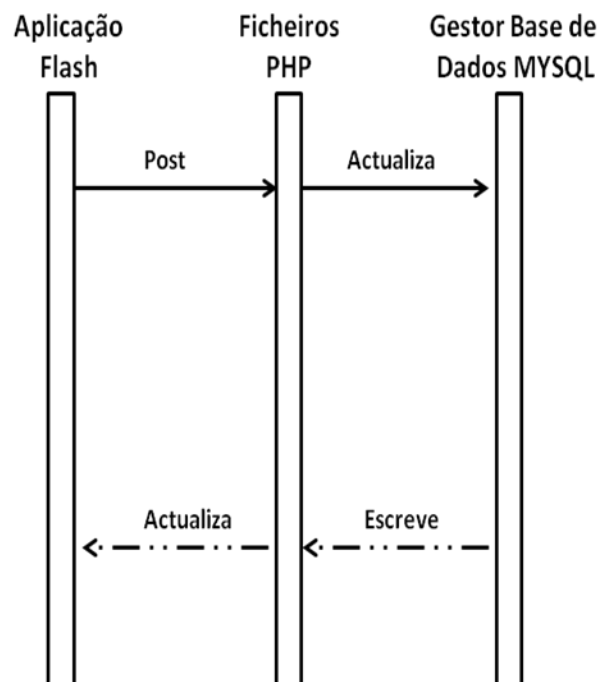


Figura 68 - Esquema de funcionamento da aplicação flash desenvolvida.



O controlo remoto dos equipamentos é efectuado na aplicação flash através de eventos associados ao pulsar dos botões do rato. A monitorização é efectuada automaticamente pela aplicação. O utilizador só tem de associar os símbolos ao estado dos equipamentos. Na Figura 69 pode observar-se a aplicação *flash* desenvolvida.



Figura 69 - Aspecto da aplicação Flash desenvolvida.

Na barra de menus de funções básicas é activada uma caixa de texto, indicativa do respectivo nome, sempre que o apontador passe sobre o ícone (Figura 70).



Figura 70 - Exemplo de navegação entre menus.

As diferenças entre a aplicação em flash desenvolvida, a interface em Visual Basic e a página em HTML e PHP são meramente a nível gráfico, uma vez que as funções efectuadas são as mesmas.

Na Figura 71 pode visualizar-se o menu de iluminação geral, onde é possível, por exemplo, apagar ou acender os equipamentos associados ao sistema de iluminação.



Figura 71 – Menu de Iluminação geral na aplicação *flash* desenvolvida.

Na Figura 72 pode visualizar-se o controlo remoto da lâmpada da sala.



Figura 72 – Menu de controlo remoto da iluminação da sala.

Na Figura 73 pode visualizar-se o menu de funções inteligentes, onde as funções de cada um dos símbolos foram já descritas anteriormente.



Figura 73 – Menu das funções inteligentes.

## 4.2 Análise de desempenho

O desenvolvimento do protótipo seguiu a arquitectura proposta descrita nos capítulos anteriores. O *software* desenvolvido dá liberdade ao utilizador de optar pelo controlo manual ou automático de um equipamento. Entende-se por controlo manual, a acção directa do utilizador sobre um equipamento numa das interfaces desenvolvidas (interface em Visual Basic, interface Web). O controlo automático é definido na tabela de configuração do protótipo desenvolvido, onde se configuram o número as entradas que condicionam o estado do equipamento pertencente ao protótipo.

A análise de resultados relaciona-se essencialmente com o aspecto de resposta temporal do sistema ao controlo efectuado por parte do utilizador ou em função da mudança do estado das entradas que controlam a saída do sistema. Observam-se tempos de resposta diferentes consoante o sistema era controlado automaticamente ou manualmente (em PHP ou na interface Visual Basic desenvolvida). Também se verificaram diferenças consoante as tecnologias envolvidas na integração de sistemas.

A Tabela 3 compara os valores para os tempos de execução em função dos sistemas integrados no protótipo desenvolvido e do modo de funcionamento do mesmo. Onde existe integração de tecnologias, a comparação de tempos foi efectuada com base no *software* desenvolvido em *Visual Basic*, desactivando-se as funções correspondentes às tecnologias. A medição dos tempos das tecnologias não integradas foi realizada com recurso a *software* fornecido com os equipamentos. Os resultados apresentados correspondem ao valor médio das medidas efectuadas.

Tabela 3 - Tempos de reacção do sistema protótipo proposto

Sistemas Integrados	Modo de Funcionamento	Software utilizado	Tempo Medido (segundos)
PC – Rede de Autómatos	Modo Automático	Visual Basic	<1,5
PC- Qbus -Autómatos	Modo Automático	Visual Basic	<2,0
PC- Qbus -X10- Autómatos	Modo Automático	Visual Basic	<2,5
PC- Qbus -X10 – CmdRF- Autómatos	Modo Automático	Visual Basic	<6,0
PC- Qbus -X10 – CmdRF- Autómatos	Modo Manual	Visual Basic	<1,5
PC – Rede de Autómatos	Modo Manual	Visual Basic	<1,0
PC - Autómatos	Modo Manual	Fatek Facon Server	<1,0
Qbus	Modo Manual	Qbus Serial Manager	<1,0
PC – X10	Modo Manual	Active Home	<2,0
PC – X10	Modo Manual	Visual Basic	<2,0

Assim, perante a análise da Tabela 3 podemos verificar que o sistema protótipo reage mais rapidamente no modo manual. O tempo de reacção do sistema aumenta nos sistemas integrados em função das tecnologias envolvidas. Quando envolvemos a tecnologia X10 no protótipo houve a necessidade de regular as funções de temporização do *software* desenvolvido, conseguindo desta forma respeitar as capacidades de resposta permitida por este protocolo.

## 5 Conclusão

O valor acrescentado deste trabalho passa pela integração de diferentes sistemas de domótica e pela interligação a diversas tecnologias de suporte existentes. Assim, neste trabalho foi implementada uma arquitectura que pressupõe o PC como elemento central. Verificou-se que este assume principal preponderância na arquitectura proposta porque permite a integração dos sistemas de domótica, o controlo e a monitorização remota, a interligação a diferentes tecnologias de suporte e oferece uma interface de utilização simples e amigável.

Conseguiu-se na arquitectura proposta integrar diversos sistemas de domótica com padrões diferenciados no modo como actuam, nomeadamente equipamentos com o protocolo X10, o protocolo Qbus e uma rede de autómatos.

No âmbito deste trabalho foi desenvolvido um conjunto de software que permite o funcionamento do sistema protótipo implementado. Relativamente ao protocolo X10, conseguiu-se desenvolver uma biblioteca de funções em *Visual Basic* que permite a funcionalidade da generalidade dos módulos pertencentes a este protocolo. Verificou-se que este protocolo é o mais lento, facto que obrigou a regular as funções de temporização do *software* desenvolvido em concordância com a capacidade de resposta permitida por este protocolo.

O *software* desenvolvido em *Visual Basic* para o controlo e monitorização no protocolo Qbus recorre à biblioteca de funções qbuscomm.dll fornecida no site do Qbus. Desta forma, foram adicionadas ao projecto as funções que permitem controlar e monitorizar os módulos de Qbus pertencentes ao protótipo desenvolvido.

Relativamente à rede de autómatos foi proposta uma estrutura de mensagens que permite a comunicação entre autómatos, e entre o autómato central e o PC. Neste contexto, foram desenvolvidas as correspondentes rotinas em linguagem *ladder* que permitem a interpretação e a elaboração da mensagem proposta.

No âmbito da interacção com o utilizador, foram desenvolvidas três interfaces de controlo e monitorização do protótipo: uma interface em *Visual Basic* que permite o controlo e monitorização do protótipo do lado do servidor, e duas páginas Web, uma em HTML e PHP e outra em *Flash*, que permitem o controlo e monitorização dos equipamentos do modelo protótipo.

A interface em Visual Basic desenvolvida revela-se eficaz nos processos de controlo e monitorização, permitindo uma rápida actuação sobre os equipamentos do modelo protótipo. Graficamente, esta apresenta um aspecto amigável e de simples utilização.

A interface em HTML e PHP desenvolvida, embora funcional, uma vez que permite efectuar o controlo e a monitorização remota dos equipamentos, não se tornava simples de usar e não apresentava um aspecto gráfico adequado à função HMI. Por outro lado, a interface com conteúdos flash apresenta um aspecto amigável a nível gráfico e permite ao utilizador uma simples interacção com o modelo protótipo.

Verificou-se que este trabalho pode ser de grande utilidade porque mostrou que é possível a interligação de sistemas de domótica com ausência de padrões normalizados, a partilha de recursos de tecnologias de suporte diferenciadas e que um PC, apesar de não ser na minha opinião a solução ideal, pode assumir a responsabilidade de gerir a integração dos diversos sistemas de domótica. Esta capacidade de interligação permitirá incrementar a expansibilidade dos sistemas de domótica e aumentar o numero de funcionalidades permitida.

No final deste trabalho, e tendo em consideração o constante aparecimento de tecnologias de suporte que permitem o controlo remoto de diversos equipamentos, considera-se ser útil e relativamente fácil de, a partir do que já foi realizado, construir uma interface que permitisse a um PDA controlar e monitorizar o protótipo desenvolvido. Este seria um dos aspectos a melhorar em futuros desenvolvimentos deste trabalho.

## 6 Referencias Bibliográficas Utilizadas

- [1] Tecnicontrol - Electrónica, Segurança e Comunicações, Lda(2005,Domótica – acedido em: 10-8-2008, em: <http://www.tecnicontrol.pt/domotica/domotica.html>
- [2] Kimaldi Electronics(2007),Produtos e sistemas autónomos - acedido em: 10-8-2008, em:[http://www.kimaldi.com/kimaldi\\_por/productos/control\\_de\\_acceso/control\\_de\\_acceso\\_autonomo](http://www.kimaldi.com/kimaldi_por/productos/control_de_acceso/control_de_acceso_autonomo)
- [3] Tecnicontrol - Electrónica, Segurança e Comunicações, Lda(2005, Sistemas de Segurança – acedido em: 10-8-2008, em: [http://www.tecnicontrol.pt/seguranca/prod\\_segur.html](http://www.tecnicontrol.pt/seguranca/prod_segur.html)
- [4] Tecnicontrol - Electrónica, Segurança e Comunicações, Lda(2005, Sistemas de Vídeo Vigilância – acedido em: 10-8-2008.
- [5] EuroX10.com(2008),EuroX10 A evolução do centralcasa.com, Acedido em:11/02/2008,em:<http://www.eurox10.com/Content/x10information.htm>
- [6] EuroX10.com(2008),EuroX10 A evolução do centralcasa.com, Acedido em:11/02/2008,em:<http://www.eurox10.com/Content/X10SignalTheory.htm>
- [7] KNX 2008 EIB – European Installation Bus - acedido em: 10-05-2008
- [8] Intelilar 2006 – Manular Técnico de Formação
- [9] ECHELON, CORPORATION. The LonWorks network plataforma – a technology overview. Disponível em: <http://echelon.com/developers/LonWorks/default.htm>. Acesso em: 6 de Out. 2008.
- [10] Ten Ways to Bulletproof RS-485 Interfaces – National semiconductors – Application Note 1067
- [11] Transceivers and Repeaters Meeting the EIA RS-485 Interface Standard – National semiconductor - Application Note 409
- [12]Wikipedia (2008), Wikipedia, A enciclopédia livre, Acedido em:11/09/2008,em: <http://pt.wikipedia.org/wiki/Tcp/ip>
- [13] CentralCasa.com(2008), CentralCasa|Tudo para Domtica|X10, Acedido em:11/10/2008,em: <http://www.centralcasa.com>
- [14] QBUS (2006),Qbus Domotica, Acedido em: 10-8-2008, em: <http://www.qbus.be/>
- [15] FATEK AUTOMATION CORP(2004), Fatek Fbs series PLC, acedido em Acedido em: 10-8-2008, em: <http://www.fatek.com>



## **Anexos**

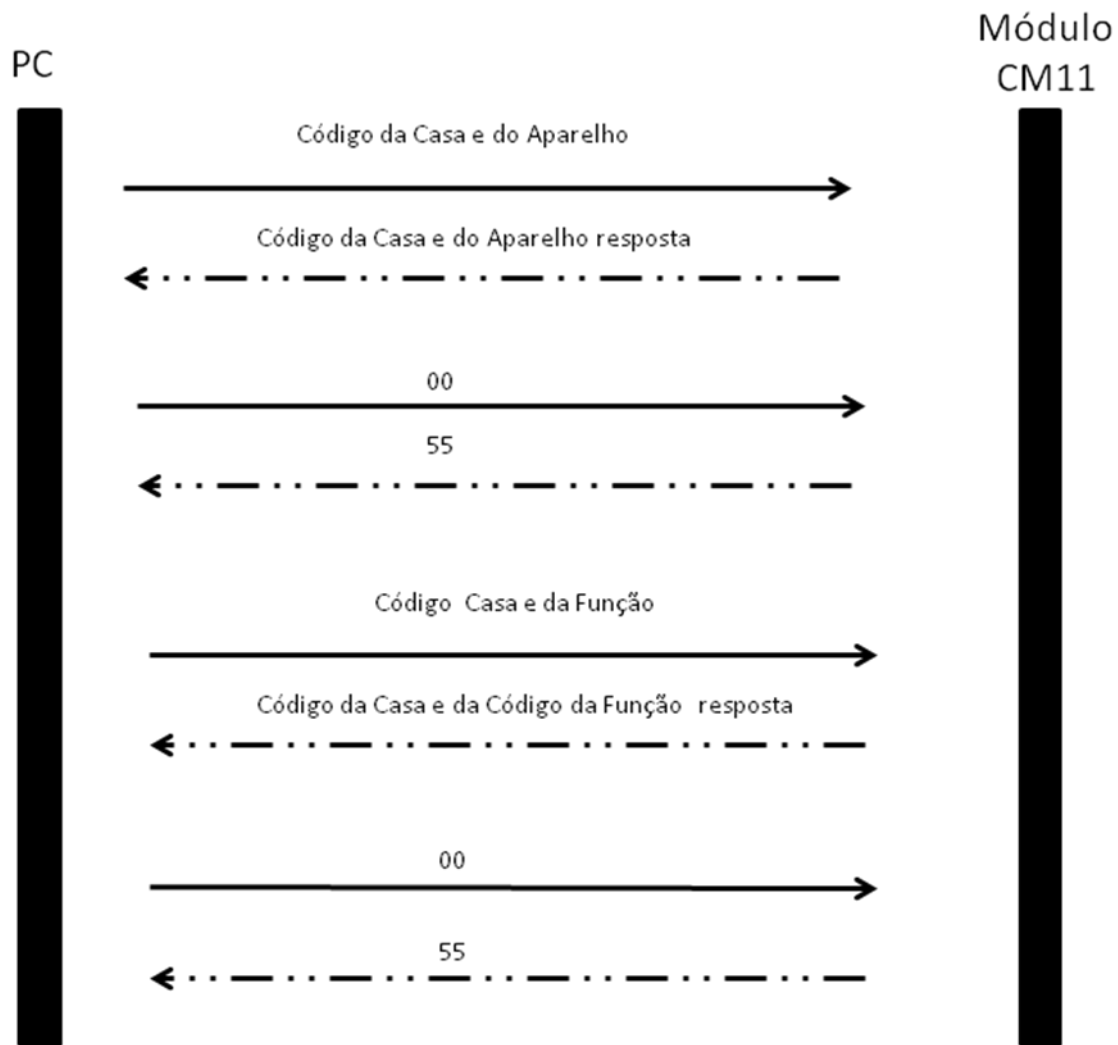
Anexo 1 – Protocolo de Comunicação entre o PC e o Módulo CM 11 no sistema X10.

Anexo 2 – Protocolo de Comunicação entre o PC e o Módulo CM 11 no sistema X10, início da comunicação.

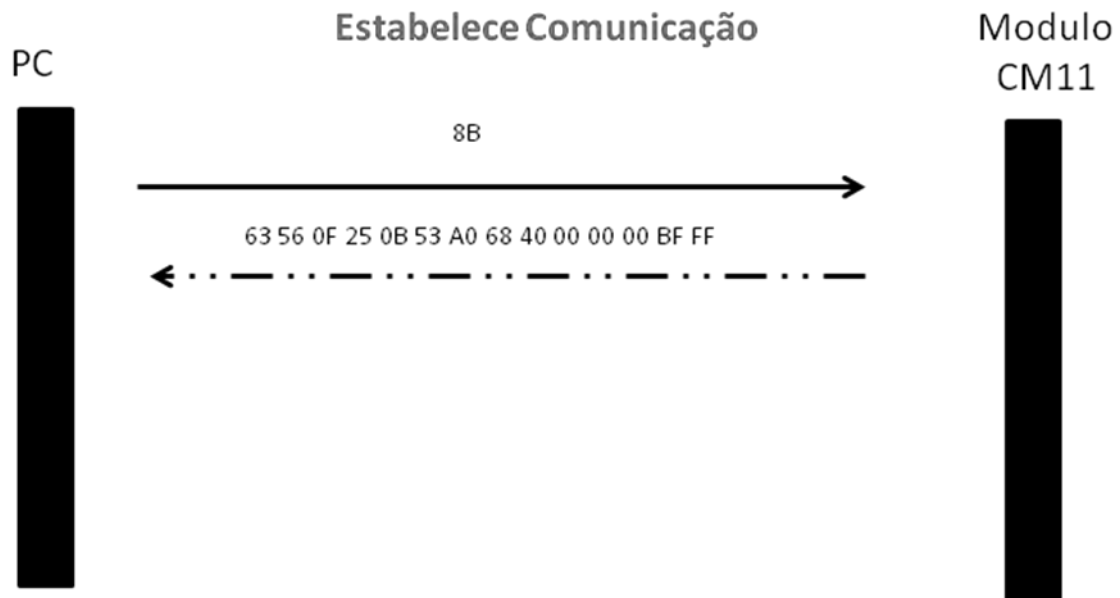
Anexo 3 - Exemplo do protocolo de comunicação entre o PC e o Módulo CM 11 no sistema X10.

Anexo 4 – Exemplo do protocolo de comunicação entre o PC e o Módulo CM 11 no sistema X10.

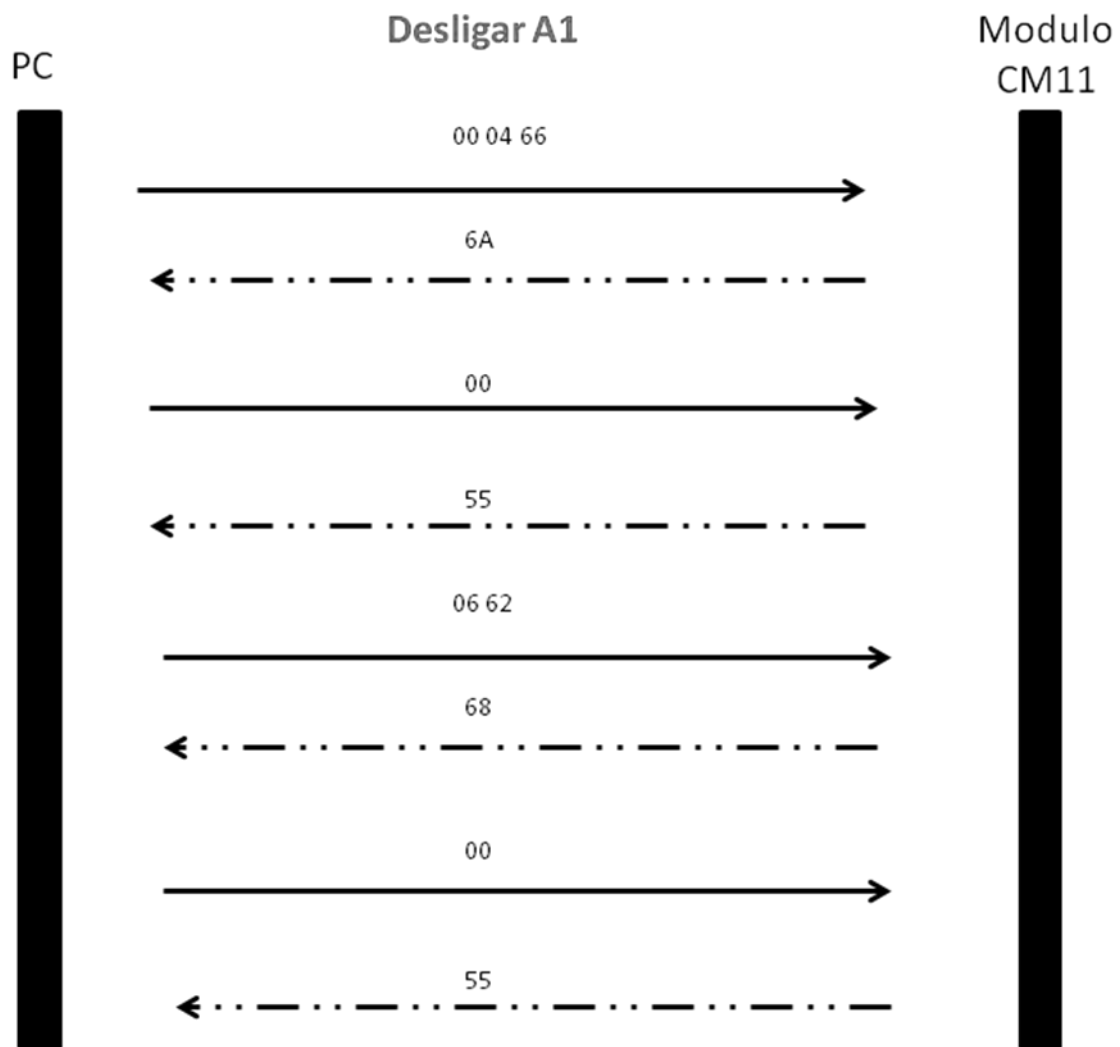
**Anexo 1** – Protocolo de Comunicação entre o PC e o Módulo CM 11 no sistema X10.



**Anexo 2** – Protocolo de Comunicação entre o PC e o Módulo CM 11 no sistema X10, início da comunicação.



**Anexo 3** - Exemplo do protocolo de comunicação entre o PC e o Módulo CM 11 no sistema X10.



**Anexo 4** – Exemplo do protocolo de comunicação entre o PC e o Módulo CM 11 no sistema X10.

